ČASOPIS ZA INFORMATIKU I RAČUNARSTVO

# računari

66
NOVEMBAR 1990.
CENA 30 D

116 STRANA

TESTOVI SOFTVERA

**ULTRASCRIPT**

**PC TOOLS 6.0**

**KWIK POWER PAK**

TESTOVI HARDVERA

**EPSON LQ 550**

**LASERPORT PS 600**

UPOREDNI TEST

**TRI TVRDA DISKA**




DODATAK NA 16 STRANA

**OBJEKTNO PROGRAMIRANJE**

CLIPPER

**KORISNIČKI INTERFEJS**

SEZAM NA 16 LINIJA I NOVOM HARDVERU

**OD SADA NA RAČUNARIMA COMPAQ**

YU ISSN 0352-7271
9 770352 727009

# računari 66

časopis za informatiku
i računarstvo
YU ISSN 0352-7221
Izdaje BIGZ
novembar 1990. cena 30

**Izdaje i štampa**
Beogradski izdavačko-grafički zavod
10000 Beograd
Bulevar Vojvode Mišića 17
**Generalni direktor**
Ilija Rapajić
**Direktor Novinskog sektora**
Zoran Nikodijević
**Glavni i odgovorni urednik**
Jovan Regasek
**Zamenik gl. i odg. urednika**
Esad Jakupović
**Tehnički urednik/sekretar redakcije**
Natalija Desančić
**Fotografije**
Aleksandar Milosavljević
**Stručna redakcija**
Vesna Ćosić (aktuelnosti), Dejan Ristanović (softver), Jovan Skuljan (programiranje), Dejan Veselinović (hardver), Zoran Životić (softver)
**Izdavački savet**
Sava Bajic, Vlado Bjelić, Drago Indić, Esad Jakupović, Roman Joakim, Dragoljub Jakac, dr Draško Milićević, dr Ljubomir Radanović, Jovan Regasek, Dejan Ristanović, dr Milivoje Simonović, Mirjana Todorović, dr Dragan Uvelić, Dragoljub Vasić
**Adresa redakcije**
11000 Beograd
Bulevar vojvode Mišića 17/III
**Telefoni**
653-748 (redakcija)
650-526 (prodaja)
**Agencija BIGZ-a**
(011) 651-793, 653-565
**Teleks**
11855 BIGZ YU
**Telefaks**
(011) 648 140
**Sezam**
(011) 647-955
**Pretplata za zemlju**
Za jednu godinu 360
Za šest meseci 180
(na žiro-račun RO BIGZ 60802-603-23264)
**Pretplata za inostranstvo**
Za jednu godinu 33 USD, 51 DEM, 43 CHS, 18 GBP, 172 FRF
(na devizni račun RO BIGZ 60811-620-16101 820701 939 03377)
Zbog usporenog prenošenja uplata preko banke, molimo pretplatnike da nam posle svake nove uplate odmah pošalju foto kopiju uplatnice
Rukopisi se ne vraćaju
**SEZAM**
sistem za modemske komunikacije
011/647 955, 24 časa dnevno
**Upravnik sisteme**
Jova Regasek
**Administratori na sistemu**
Zoran Životić i Dejan Ristanović
**Softver**
Zoran Životić (glavne aplikacija) i Dejan Ristanović (eksterni programi)
**Hardver**
Server COMPAQ DESKPRO 386/25, 1 MB RAM-a, tvrdi disk 300 MB (COMPAQ Minhen). Mreža: dve stanice COMPAQ 286N/12, 1 MB RAM-a, tvrdi disk 40 MB (COMPAQ Minhen), tri stanice "Ultra" AT 12 MHz, D-LINK DE 100 LanSmart ("Ultra" Skopje). Modemi ViVa (Theleme Communitations Beograd). Razvoj višelinijskog "Sezama" značajno pomogli "Gama" i "Memodata", Beograd

## SADRŽAJ



**UMETAK**

**na 16 strana**

**Duška Savić**

**OBJEKTNO PROGRAMIRANJE**

**NA TURBO PASCALU 5.5**

**so kompletnim programom spremnim za upotrebu**

ŠONSI

# ZOVE NAS CEO SVET

**Gens una sumus (Svi smo jedan svet)! Ovih dana ceo svet se preselio u Novi Sad – počela je 29. šahovska olimpijada. Jedan ozbiljan deo tog sveta – šahovske federacije, novinske organizacije, šahisti, ljubitelji šaha – čini to posredstvom našeg Sezama, koji funkcioniše kao službeni informacioni sistem olimpijade. Sezam se, dakle, otvorio za ceo svet.**

Šah nadamo se volite? Telefon pretpostavljamo imate? Možda imate i računar? A modem? Ako je odgovor na sva ova pitanja potvrdan onda ste i vi jedan od domaćih srećnika kojima Sezam, na broju 011/64 88-99 – olimpijadu nudi kao na dlanu mogućnost da u svako doba pregledate i presnimite bilo koju odigranu partiju, računajući i one koje su još „vruće", kao i tabele liste i druge tekuće informacije priliku da i sami komentarišete debatujete ili na bilo koji drugi način reagujete na ono što se događa na olimpijadi i oko nje šansu da sa šahovskim svetom razmenjujete javnu ili privatnu poštu

### MOTIVACIJA I OPTIMIZAM

Premda modemske komunikacije više i nisu neka novost ovakav model praćenja šahovske olimpijade – ili bilo kojih olimpijskih igara – jeste nov i prvi put se „proverava u praksi" S razlogom verujemo da će Sezamova ili slične koncepcija sistema za praćenje velikih sportskih manifestacija od sada pa ubuduće biti pravilo a ne izuzetak Drugo je pitanje što kod nas još uvek moramo da govorimo o „srećnicima" koji imaju mogućnost da iskoriste ono što Sezam (ili bilo koji BBS) nudi za olimpijadu ili nezavisno od nje Na Zapadu su modemske komunikacije „obična stvar" – što je je l da strašno čudno s obzirom da je računarska oprema tamo samo dvaput jeftinija a zarade su tek pet do deset puta veće

Prema stotinama hiljada korisnika modemskih komunikacija u razvijenim zemljama, domaće „modemske snage" možda deluju kao „mačji kašalj" u upotrebi je tek nekoliko hiljada modema Pa ipak na čitavu stvar se mora gledati sa više vedrine Treba imati na umu da se na tlu (federativno/konfederativne) Jugoslavije – prema „statistici" u koju se ne možemo previše pouzdati pošto zvanični podaci o tome ne postoje, ali koja je ipak zasnovana na iskustvenoj proceni (takozvano „merodavno mišljenje") – koristi preko tri stotine hiljada personalnih računara pretežno u privatnim rukama S obzirom da u svetu prosečna cena modema (reda 200 DEM) ipak iznosi samo deseti deo prosečne cene PC računara (reda recimo 2000 DEM) prelazak na modemsko komuniciranje i ne predstavlja neki naročito težak finansijski poduhvat Problem je, zapravo motivacija – a na tom planu naša vlastita iskustva pružaju dosta optimizma

Dobar deo od onih nekoliko hiljada vlasnika modema u našoj zemlji naime makar povremeno bira Sezamov broj telefona (od sada će naravno veza biti dostupna samo pretplatnicima čiji broj ovog trenutka iznosi nekoliko stotina i svakodnevno se povećava) Važnije je međutim što je veliki broj korisnika izuzetno aktivan – mnogi među njima prisutni su praktič

### SEZAM U SLUŽBI OLIMPIJADE

*Sedište Sezama*
- BIGZ – redakcija „Računara"

Beograd Bul vojvode Mišića 17

*Radno vreme*
- non stop

*Osnovni deo*
- 3 linije za pretplatnike (broj 011/647-955)
- 1 linija za održavanje
- jezici srpskohrvatski,slovenački,engleski
- godišnja pretplata 800.00 dinara

*"Olimpijski" deo*
- 2 linije za neprekidnu vezu sa N Sadom
- 10 linija za šahovski svet

**(broj 0038/11/648-899)**
- jezik engleski
- korišćenje besplatno

*Hardver*
- Server Compaq 386/25
- Stanice Compaq Ultra Micronic
- Mreža Schnaider & Koch
- Modemi ViVa Micronic

*Softver*
- Glavna aplikacija Zoran Živolić
- Eksterni programi Dejan Ristanović

**Upravnik sistema**
Jova Regasek

**Administratori na sistemu**
Zoran Živolić i Dejan Ristanović

*Informacije o Sezamu*
Redakcija „Računari"
- tel 011/653-746 • fax 011/648-14

### LIČNA KARTA OLIMPIJADE

- Dvadeset deveta šahovska olimpijada je zapravo 29. muška i 14. ženska šahovska olimpijada.
- Održava se od 16. novembra do 4. decembra u Sportskom i poslovnom centru „Vojvodina" u Novom Sadu.
- Svečano otvaranje 16 novembra u 20 00 sati
- U muškoj konkurenciji učestvuje 108 timova iz 107 zemalja, a u ženskoj 56 ekipa iz 55 zemalja, što predstavlja rekord, naročito ako se ima u vidu broj igrača.
- Redovna kola se igraju 17- 21.11, 23-27.11 29- 30 11, 1 12 i 3 12 od 15.00 do 21 00 (nastavci 10 00- 12 00)
- Zatvaranje i dodela nagrada 4. decembra u 11 00
- Za vreme olimpijade održava se i 61. kongres FIDA (Generalna skupština 30 11 –2 12)
- Održavaju se i tri otvorena turnira: 9-17.11, 18- 26 11 i 27 11 - 5 12.
- Svakog dana 17 11 - 3.12. održavaju se brzopotezni turniri.
- Turnir u rešavanju šahovskih problema 28 novembra
- Planiran je niz pratećih aktivnosti turnir šahovskih kompjutera, šahovske izložbe (karikatura umetnička fotografija kolekcionarstvo istorija YU šaha), prodaja i razmena šahovske literature, programa, video kaseta rekvizita i opreme



Foto Aleksandar Milosavljević

Svečana predaja COMPAQ ovih mašina u redakciji „Računara" Predstavnik firme COMPAQ gospodin Matijas Rajkaj (Mathias Rajkay) i generalni direktor BIGZ-a gospodin Ilija Rapaic

# SEZAM NA RAČUNARIMA COMPAQ

Za nešto više od godinu dana svoga postojanja „Sezam" je u skladu sa rastom interesovanja i broja članova nekoliko puta menjao svoj i softverski i hardverski lik. Zahvaljujući donaciji firme COMPAQ kao i zagrebečkom MicroLAB u koji je na sebe preuzeo carinske formalnosti, „Sezam" je početkom novembre prešao sa komercijalne na profesionalnu klasu hardvera. Za korisnike to, pre svega, znači lakši, brži i pouzdaniji put do informacije koje su mu potrebne, a za redakciju „Računara" i upravu „Sezama" izuzetno priznanje i – znatno manje glavobolja u održavanju sistema.

Kompletan „Sezam" sada pokreće COMPAQ DESKPRO 386 na 25 MHz sa procesorskim kešom od 32 K i tvrdim diskom kapaciteta 300 megabajta. Ova mašina je specijalno projektovana za pogon mreža i očekujemo da će sa lakoćom da opslužuje 16 „Sezamovih" nodova. Dosadašnji server jedan AT na 12 MHz koji nam je isporučile „Gama" odlazi u zasluženu penziju, na mirnije redakcijske zadatke.

Za vreme Olimpijade svi podaci će se sliveti u „Sezam" preko dve specijalne stanice koje će preko tzv. poprečnih veza biti u stalnoj vezi sa Novim Sadom. Na ovim poslovima će raditi dve COMPAQ stanice 286N na 12 MHz sa tvrdim diskom kapaciteta 40 MB, takođe specijalno projektovane za rad u mrežama. Dobar deo poslova oko komunikacije sa komandnim centrom Olimpijade – prikupljenje i obrada rezultata i – ove stanice će obavljati samostalno, ostavljajući serveru više vremena za komunikaciju sa korisnicima iz celog sveta.

Standardne "Sezamove" stranice koje imaju najviši prioritet s prema tome trpe najveće opterećenje zadržale su i u ovoj prilici svoju ulogu. Radi se o tri potpuno nove „Ultra" AT mrežne stanice na 12 MHz bez tvrdih diskova („Ultra" Skopje). Preko ovih stanice na poznatom broju 011 647-955 dobijaće se uobičajene „Sezamove" usluge. U standardnu postavu spada i nod za održavanje sisteme. Preko „Riopline sx" na 16 MHz, sa tvrdim diskom od 40 MB (DVEnterpises Beograd), na posebnom broju da nikome ne oduzimaju vreme administratori i moderatori diskretno upravljaju sistemom. Sa ovim delom „Sezama" komunicira se preko modema „Vi-Ve" (Thelema Communications, Beograd).



**Novi server COMPAQ 386 na 25 MHz sa procesorskim kešom od 32 K i tvrdim diskom kapaciteta 300 megabajta**

Za potrebe Olimpijade uvodimo desat posebnih linija na broju 011 648 889. Računare – XT mašine sa procesorom 286 – i modeme – za ovu priliku je obezbedio „Micronic" (Zagreb), firma koja se proslavila svojom računarskom podrškom za poznale televizijske emisije ("Kviskoteka" i mnoge druge). Preko ovih stanica dolaziće pozivi iz celog sveta.

"Sezam" standardno radi u D Link mreži pod operativnim sistemom *SmartLink*. Za vreme Olimpijade sistem privremeno prelazi na *Novell*, a računari će biti umreženi preko profesionalnih Ethernet mrežnih kartica Schneider & Koch. Kompletnu softversku i harversku instalaciju mreže je obezbedio MicroLAB (Zagreb), ekskluzivni zastupnik ovog poznatog svetskog proizvođača.

Iz jednog kloniranog AT-a koji je u međuvremenu potpuno potrošen, „Sezam" je za samo godinu dana izrastao u glomazan sistem za modemske komunikacije koji pokreće najkvalitetnija softverska a sada i hardverska oprema. U opremanju „Sezama" udružile su se firme sa čitavog jugoslovenskog prostore – iz Beograda, Zagreba i Skopja – i, što je od posebnog značaja COMPAQ, proizvođač računara svetskog renomea, koji donacijom iza „Sezama" staje ne samo svojim pouzdanim hardverom nego i svojim – imenom.

## „MICRONIC" RAČUNARI NA OLIMPIJADI

Informacioni sistem ŠONS-a podržan je velikim brojem računara, smeštenim u sportskoj palati, Pres-centru, dvorani za Kongres FIDA, i u objektima za smeštaj igrača, sudija i novinara – koji stalno prate rezultate partija i mečeva, stanje tabela, tok svake šahovske partije.

Ovu mnogobrojnu računarsku opremu obezbedio je MICRONIC COMPUTERS, kompjuterska kuća iz Zagreba, već preko dvanaest godina prisutna na domaćem i stranom tržištu, poznata po kvalitetnim proizvodima, dobrim servisima na čitavoj teritoriji Jugoslavije, te odličnom softveru svoje softverske kuće „Microcode" iz Novo Sada.

„Micronic" se pročuo po postavljanju kompletne obrade za tadašnje regionalne SIZ-ove, zatim informatizacii naših lutrijskih organizacija, pa razvoju trgovačke kompjuterske kase, potom razvoju džinovskog automatizovanog informacionog sistema DP Plitvice sa 26 lokacija, projektu prenosivog sistema kontrole proizvodnje pa projektu totalne informatičke podrške za sisteme robnih kuća.

„Micronic" danas obezbeđuje kompletan inženjering koji se oslanja na UNIX TCP/IP MICRONIC POS sisteme, široki spektar uređaja za akviziciju podataka te proverenu i brzu softversku podršku.

Svi MICRONIC računari zasnivaju se na procesoru INTEL 8086 Olivetti LSX i CP 486 NOVELL i UMD mreže, odnosno pod operativnim sistemima i operativnim okolinama MS-DOS MOS CP/M CCPM CDOS UNIX XENIH H/OS TCP/IP itd).

no na svim konferencijama intenzivno razmenjuju privatnu i javnu poštu, međusobno se informišu o novostima, daju savete. Oni, dakle, imaju motiv komunicirаju jedni s drugima, razmenjuju informacije, iskustva, znanje, zadovoljavaju svoju potrebu da raspravljaju o aktuelnim događajima iz sveta računarstva, politike, kulture...

### VIŠE OD „SEZANJA"

Modemski sistem, slično kao i sâma telefonska mreža, ima smisla tek ako korisnike ne skuplja – kako je jednom slikovito rekao jedan urednik jedne ozbiljne a netiražne edicije – kao filatelisti marke dakle lupom i pincetama. Tri telefona u mreži i nisu neka dobra prilika za ćaskanje i obavljanje poslova. Ali tri stotine... A tek tri hiljade... I sad kakva je budućnost domaćih BBS-ova uopšte, a naročito na-

šeg Sezama (kao bez sumnje vodećeg)? Od govori je kratak: ekspanzija! „Pokret" sezamovaca se neumitno širi po sistemu „više korisnika – više i koristi". U ovih samo godinu dana svog postojanja Sezam je od neobaveznog sredstva za konferansije, informisanje i „sezanje" – a zadržavajući sve to – postao i jedan krajnje ozbiljan informacioni sistem za jedan događaj od globalnog značaja.

Pripreme za praćenje 29. šahovske olimpijade u Novom Sadu (ŠONS) bile su prave „porođajne": prvo – trebalo je razviti novi softver, obezbediti dodatnih dvanaest telefonskih linija, srediti telefonske i električne instalacije, pripremiti prostor; drugo – nabaviti novi server, „luce" računare i drugi hardver, oformiti novu mrežu; treće – razrešiti gomilu organizacionih i administrativnih problema; četvrto – preživeti konferencije za štampu, učestale posete kolega sa televizije... U prvom (i četvrtom) smo se oslonili na svoje „standardne" snage (redakcija i sysop-i), u drugom su u pomoć priskočili Compaq/MicroLab i drugi (videti prilog), kod trećeg su se pri ruci našli ljudi iz BIGZ-a i našeg Novinskog sektora. U većini ovih poslova, razume se, Organizacioni komitet ŠONS-a poneo je svoj deo tereta. Održavanje sistema za vreme olimpijade pomalo mu dođe kao odmor posle čitave te trke, mada napetost neće popustiti dok se sve ne završi.

Šta Sezam pruža šahovskom svetu? Pre svega „klasične" informacije o toku takmičenja (tabele, razne prateće liste, zanimljivosti) kakve inače nude i drugi mediji, ali naš BBS to čini neuporedivo većom brzinom, uz mogućnost direktnog preuzimanja u obliku praktično poolpuno sređenom za dalje korišćenje čak i bez prekucavanja.

## NIŠTA BEZ „SEZAMA"

Zatim, naš „četvrti medij" korisnicima nudi dve mogućnosti koje tri klasična medija (štampa, radio, televizija) ne mogu da im pruže:

- **NEPOSREDAN UVID U SVE ODIGRANE PARTIJE.** Pošto pozove olimpijsku konferenciju Sezama korisnik dobija mogućnost da besplatno pregleda listinge ranijih i tek odigranih partija, odabere one koje ga zanimaju, ili sve, i prenese ih u svoj računar da bi ih kasnije natenane gledao i analizirao. I
- **DIREKTAN KONTAKT SA UČESNICIMA OLIMPIJADE.** Svaki korisnik stiče priliku da razmenjuju javne poruke, a posredstvom elektronske pošte i privatne poruke sa učesnicima i organizatorima olimpijade – da, na primer, komentariše pojedine partije ili zanimljive događaje, predlaže poteze u prekinutim partijama, kao i da šalje i prima sasvim lične poruke.

Sezam se, dakle, ovih dana nalazi u službi jedne svetske manifestacije. Ali, bez ikakve sumnje, i olimpijada je na neki način „u službi" Sezama, afirmišući ga kao jedan ozbiljan i višestruko konstan komunikacioni sistem čije vreme sve više dolazi, omogućujući mu da se potvrdi u našoj zemlji ali i u svetskim relacijama, i ostavljajući mu u amanet veliku bazu podataka dragocenu za poklonike šaha, kralja svih igara.

Naš BBS je sada na svojoj velikoj probi na kojoj će, nema razloga da sumnjamo, dokazati svoje kvalitete. Posle olimpijade Sezam više neće biti isti, jer će iza njega stajati jedno značajno iskustvo visokih dometa. Verujemo tako de da ni olimpijada više neće biti ista – zato što će se potvrditi da modemske komunikacije ovakvim svetskim manifestacijama mogu da pruže nezamenljive kvalitete bez kojih se više jednostavno neće moći: nenadmašnu brzinu, dinamičku informaciju (koja se stalno „ažurira"), globalni domet, interaktivan odnos.

S obzirom da olimpijade i slične svetske manifestacije prati snažan marketing, od velikog je značaja i činjenica da Sezam (BBS uopšte) i u tom pogledu pruža nekoliko potpuno novih oblika tržišnih komunikacija kakve u klasičnim glasilima naprosto nisu moguće (opširniju informaciju o tome naći ćete u ovom broju „Računara" na strani 67).

Ovih dana u svojim reklamama lansirali smo slogan: „Zove nas ceo svet – javite se i vi!"

Stvarno se javite, ne samo dok traje olimpijada.

*Esad Jakupović*

## INFORMACIONI SISTEM

U nedavno objavljenom Biltenu br. 3 olimpijade naglašava se da informacioni sistem ŠONS-a – o čijem radu brine poseban tim – jedinstven po vrstama različitih poslova i količini informacija koje se obrađuju, obezbeđuje specifična rešenja, široku tehničku i tehnološku osnovu i specifične aplikacije. Razvijena su četiri informaciona podsistema: akreditacija; parovanje šahovskih mečeva; redakcija Biltena; Distribucija informacija.

U vezi sa ovim poslednjim segmentom, u Biltenu se kaže sledeće: „Ključnu ulogu celog informacionog sistema ima Sezam – koji će šahistima širom sveta, bez obzira gde se nalazili, omogućiti pomoć koju nijedan klasični informativni medij ne može da pruži." Čitava stvar se zatim detaljnije obrazlaže.

Tekst o informacionom sistemu olimpijade, inače, nosi podnaslov „Prvi put u istoriji sporta – neposredan uvid svih zainteresovanih iz bilo kog kraja sveta u tok takmičenja".

U direktorskoj ekipi evropskog COMPAQ-a sedi i jedan Jugosloven, gospodin Želimir Ilić, direktor za razvoj tržišta, prodaju, servis i tehničku podršku u preko dvadeset zemalja Evrope, srednjeg istoka i Afrike. Gospodin Ilić je uveo COMPAQ – a sa njim, uz slavno ime, i vrhunsku tehnologiju – u desetak zemalja Evrope. Među njima, u maju ove godine, i u Jugoslaviju. Naš telefonski poziv zatekao ga je u pripremama za put u Moskvu. To je prva sledeća stanica u COMPAQ-ovom osvajanju istočne Evrope ili, kako gospodin Ilić više voli da kaže, „zemalja koje su u prelasku na tržišnu privredu".


**Želimir Ilić**

U kraćoj izjavi za „Računare" o motivima koji su naveli jednog od najuglednijih svetskih proizvođača računara da izuzetno vrednom donacijom podrži „Računare", njihov projekat „Sezam" i „Sezamovo" učešće na 29. šahovskoj omipijadi u Novom Sadu, gospodin Ilić je rekao:

*– COMPAQ sa zadovoljstvom podržava sportske takmičenja od svetskog značaja, posebno čiste, plemenite individualne sportove kao što su tenis i šah. Ove godine COMPAQ je sponzor „Grand Slam Cup" turnira u Minhenu na kome nastupa i Jugosloven Ivanišević. To je jedan od pet najvećih teniskih događaja na svetu. Šahovska olimpijada u Novom Sadu je drugi po značaju međunarodni sportski događaj koji će COMPAQ podržati u ovoj godini.*

*Velike sportske priredbe su za nas izuzetno važne u osvajanju tržišta – one omogućuju promociju COMPAQ imena pred milionima ljudi iz čitavog sveta. Informacioni sistem na turniru nacija u čijem se središtu nalazi „Sezam" obezbeđuje potpuno novi kvalitet informisanja, ali i novi kvalitet tržišnih komunikacija. Za novinare, profesionalne šahiste i ljubitelje šaha širom sveta COMPAQ neće biti samo ime sa panoa. Preko modemskog sistema „Sezam", u čijem se srcu sada nalaze COMPAQ računari, oni će, uzimanjem ključnih informacija koje ih zanimaju, moći da dobiju i naposredan osećaj rada sa COMPAQ mašinama i uvere se u njihovu brzinu, snagu i pouzdanost. Prvi put u istoriji sportskih takmičenja računari i računarske mreže se koriste u sistemu informisanja na ovako kvalitetan način.*

*Turnir nacija u Novom Sadu je za COMPAQ samo povod da se prihvati sponzorstva nad „Sezamom". COMPAQ ima dobre odnose sa računarskom štampom u čitavom svetu. Želimo na taj način da podržimo razvoj „Računara" i „Sezama", jednog modernog sredstva komuniciranja, na tržištu sa koga imamo lep odziv i do koga nam je veoma stalo. Verujemo da će računari COMPAQ svojom pouzdanošću pomoći „Sezamu" da takođe ostvari svoje ciljeve. To je, međutim, i lepa prilika za COMPAQ da na javnoj sceni, pred svetskom i jugoslovenskom publikom, pokaže ono što naši korisnici inače znaju – izuzetnu izdržljivost i pouzdanost COMPAQ mašina i na najtežim zadacima u intenzivnom neprekidnom radu 24 časa dnevno 365 dana u godini.*

## Mešani mediji

### MULTIMEDIJALNA BAZA PODATAKA

Firma HSC, poznata po grafičkom paketu *Graphics Link Plus*, na ovogodišnjem sajmu *Presentations 90* (Long Beach, California) predstavila je novu bazu podataka koja radi sa slikama, video i audio podacima i tekstom.

*SanteFe Media Manager*, kako je nazvana nova baza, predstavlja prvi paket iz serije koju ova kompanija planira da predstavi tokom sledećih nekoliko meseci, a ono što će biti zajedničko za celu seriju je novi grafički korisnički interfejs. Kao razlog za razvoj sopstvenog grafičkog interfejsa je navedena, pre svega, sporost i glomaznost *Windows*-a.

Možda najveći prodajni adut *Media Manager*-a će biti podatak da je kao osnovna baza iskorišćen Borlandov *Paradox Engine*. To znači da novi paket može da radi direktno sa postojećim *Paradox* bazama, da je kompatibilan sa *Paradox* 3.5 i da se može koristiti kao radna sredina za sve vodeće baze podataka.

*Media Manager* podržava PCX, GIF i FLI (standardni format Autodesk Animator-a) grafičke formate. Slike se mogu pregledati na četvrtini ekrana ili na punoj veličini, istovremeno slušajući zvuk povezan sa pojedinim slikama. Omogućeno je nezavisno tekstualno, brojčano, grafičko ili audio pretraživanje.

Na raspolaganju je i poseban prezentacioni mod koji, s obzirom da je program baziran na *Paradox*-u, omogućava stvari kao što je traženje svih podataka koji odgovaraju postavljenim kriterijumima i njihovo prikazivanje u obliku prezentacije.

Minimalna potrebna konfiguracija obuhvata 12 MHz AT sa 640K osnovne i 400K *expanded* memorije, super VGA monitor i karticu sa 512K RAM-a. *Media Manager* radi sa 8-bitnom VGA grafikom sve do 1024×768×256 kao i sa 16 i 24 Targa grafikom. Targa verzije zahtevaju 640 K (Targa 16) ili 960K *expanded* memorije (Targa 24). Uz to je potreban hard disk od najmanje 20 MB sa bar 5 MB slobodnog prostora, ali za velike video i audio baze je potrebno imati diskove kapaciteta između 100 i 600 MB. Od ostalih jedinica masivne memorije se mogu koristiti DAT i optički drajvovi. *Media Manager* podržava JPEG (16:1) i MPEG kompresiju, a zvučni fajlovi zahtevaju i odgovarajuću audio karticu.

Cena je, naravno, prava sitnica: pun workstation/server sistem košta $895, a svaki LAN čvor dodatnih $150.

Kontakt adresa: HSC Software, 1661 Lincoln Blvd, Suite 101, Santa Monica, CA 90404, USA. Tel. (213) 392 8441.

## Portabli

### NOVI SX LAPTOP MODELI

Modern Computer Corp., tajvanska firma koja pokušava da se probije na američko tržište, prikazala je novi 16 MHz 386SX laptop (2.8kg) nazvan *Walkom NP 903*. Za $5000 dobija se 1 MB RAM-a, 3.5" flopi, opcioni 20 ili 40 MB hard disk i 640 x 480 VGA LCD.

Novost predstavlja par ekspanzionih slotova na gornjem delu mašine koji primaju bilo koju kombinaciju od 15 modula iste firme. Postojeći moduli obuhvataju 2400 bps modem, fax emulaciju terminala, LAN interfejs, IEEE 4888 interfejs, memorijsko proširenje, dodatne baterije, dodatnu masovnu memoriju i *voice processing* modul.

Firma TOPPCs je predstavila dva nova portabl SX modela. Svaki je težak 7.3 kg i može da radi oko 2 do 3 sata bez punjenja. Mašine su po svemu, osim monitora, identične – obe su bazirane na 16 MHz 386SX procesorima, imaju 1.4 MB RAM-a, 3.5" flopi, 40MB hard disk i 640 x 480 VGA grafiku sa 16 nijansi sivog.

Model LT5600 je snabdeven supertvist LCD monitorom *paper white* tipa i košta $3700, dok je model LT5300 snabdeven amber gas-plazma monitorom i košta $4000.

Kontakt adrese: Modern Computer Corp., Rm618, Chia Hsin Bldg, 6/F, 96 Chung Shan North Rd., Sec. 2, Taipei, Taiwan, ROC, tel. 886-2-5511677, fax 886-2-5415112. TOPPCs International, 648 N. Eckhoff St., Orange, CA 92668, USA, tel. (714) 939 1416, fax (714) 939 0103.

## Čipovi

### AT MOTHERCHIP

Teksaška firma Advanced Micro Devices je razvila čip koji sadrži 80286 procesor i sve prateće čipove potrebne za izgradnju osnovne AT ploče. Novi čip je nazvan *AT Motherchip* po analogiji sa izrazom *Motherboard* – osnovna ploča. Čip je napravljen u dve verzije – 286ZX i niža verzija 286LX, objedinjuju sve funkcije procesora i pratećih komponenti potrebnih da taj procesor pretvore u računar. Umesto konstruisanja prenatrpanih osnovnih ploča koje sadrže i pedesetak čipova, sa novim čipom je za izgradnju operativnog računara dovoljno imati DRAM, kontroler tastature i sistemski bas.

Set čipova potrebnih za klasičan AT sistem pored samog procesora tipično obuhvata DMA kontrolere, interapt kontrolere, kauntere i tajmere, časovnike realnog vremena, nešto CMOS statičkih RAM ova, klok generator za sistemsko vreme, bas kontroler, DRAM interfejs i memorijske kontrolere. Novi čip ujedinjuje sve te funkcije u istom kućištu. Verzija 286ZX je predviđena za stone mašine, a 286LX za laptop i ručne računare, i rade na 12.5 i 16 MHz.

U firmi kažu da će „uzorci novih čipova uskoro biti na raspolaganju", ali odbijaju da otkriju koji proizvođači planiraju da ga koriste. Po svemu sudeći, to će biti svi značajniji proizvođači laptop i ručnih računara. Međutim, cene se znaju već sada – verzija 286ZX će koštati $69 (12 MHz) i $85.50 (16 MHz), a cene verzije 286LX će biti $76.50 (12 MHz) i $89 (16 MHz). Sve cene se odnose na količine od preko 1000 komada.

## Lokalne mreže

### PRINTERSKA MREŽA

Za korisnike koji žele da koriste jedan printer sa više radnih mesta a nisu im potrebne ostale funkcije LAN mreže, jeftine printerske mreže, spulere, predstavljaju dobru alternativu. Jedan od takvih spulera je i ESI 2089A, proizvod firme Extended Systems, koji podržava 8 ulaznih i dva izlazna uređaja i raspolaže brzom serijskom komunikacionom mrežom iste firme, Extended Link.

Extended Link kombinuje brzinu paralelnog prenosa i nižu cenu i veću daljinu prenosa serijskog prenosa, uz brzinu od 23K u sekundi i daljinu od 150 metara. Princip rada se sastoji u konvertovanju paralelnih signala u serijske na ulazu u mrežu i iz serijskih ponovo u paralelne na izlazu. Na paralelni port ulaznog uređaja se priključi adapter koji multipleksira paralelni signal u RS-422 serijski tok, koji se zatim šalje relativno jeftinim RJ-11 ili RJ-45 serijskim kablovima to ShareSpool uređaja, gde se ponovo konvertuje u paralelni signal i prosleđuje na štampač ili ploter.

2089A prihvata bilo kakvu kombinaciju od 8 serijskih (9600 bps ili 19.2 Kbps) ili Extended Link serijsko-paralelnih ulaza. Svi kanali mogu istovremeno da primaju podatke koji se smeštaju u 1 MB bafer. Na izlazu sa nalaze dva printerska porta – jedan je 19.2 Kbps serijski, a drugi 40 Kbps paralelni, što omogućave vezivanje dva različita izlazna uređaja, recimo laserski printer i serijski ploter. Oba izlazna uređaja su dostupna svim ulaznim uređajima, ali ipak ne istovremeno. Korisnicima koji žele istovremeni pristup na oba izlazna uređaja stoji na raspolaganju 2289A MultiSpool, koji je u svemu ostalom identičan SharSpool-u.

ESI 2089A ShareSpool košta $945, a 2289A MultiSpool $995, ali to nije konačna cena – svaki računar mora biti opremljen Extended Link adapterom ($60), a cena kablova iznosi $25 za 50 stopa ($1.6 po metru). Cena kompletne mreže sa svih osam priključenih ulaza i prosečnom dužinom veze od 30 m iznosi oko $1875.

Kontakt adresa: Extended Systems, 6123 N. Meeker Ave, Boise, ID 83704, USA, tel. (208) 322 7575, fax (208) 377-1906.

## Fax kartice

### SATISFAXTION

Intelova fax kartica Connection Coprocessor je dobila naslednika – nova kartica, nazvana SatisFAXtion, sadrži 9600 bps fax modem koji može i da šalje i da prima, a tu su i MNP level 5 modem od 2400 bps i port za skener.

Kartica koristi 16 MHz 80186 procesor i ima 128K memorije za pozadinski rad. Uz SatisFAXtion se isporučuje i kontrolni softver za rad pod DOS-om, a kupci za sada dobijaju i kupon za besplatni primerak programa *FAXit for Windows* koji omogućava rad iz *Windows 3.0* aplikacija. Intel takođe daje i kupon za besplatni primerak *PC Tools V.6*, koji takođe uključuje softver za rad sa SatisFAXtion.

Ploča radi tako što korisnik, pošto je pripremio materijal za slanje, iz bilo koje aplikacije izdaje naredbu za štampanje na Epsonu FX 85. Tada na scenu stupa TSR program (DOS verzija) ili FAXit program (*Windows* verzija) i preusmerava izlaz na faks umesto na printer. Podaci se šalju u obliku koji bi imali da su zaista odštampani na printeru, a korisnik pri tome ne mora da napušta aplikaciju. TSR program koristi *expanded* memoriju i okupira samo 13K regularne memorije.

Primanje telefaksa se takođe obavlja u pozadini. Program zadužen za kontrolu faks kartice jednostavno obaveštava korisnika pop up dijalogom ili zvučnim signalom i omogućava korisniku da izabere pregled ili snimanje faksa na disk za kasniju upotrebu. U paketu se nalazi još jedan program koji kontroliše mašinu u odsustvu korisnika i sve pristigle telefakse automatski štampa, kao što radi klasična telefaks mašina.

Ploča raspolaže i portom za ručni skener, čime se korisniku omogućava da dokumente koje inače ne bi mogao da proizvede direktno na računaru skenira direktno na faks ili spoji sa drugim dokumentima. Intel PCEO kupcima paketa nudi odgovarajući ručni skener (Logitech) za $399.

SatisFAXtion ploča podržava Intelov CAS programski interfejs za komunikacije, što proizvođačima softvera daje mogućnost za ugrađivanje podrške za SatisFAXtion direktno u svoje programske pakete.

Cena paketa je $499, registrovani korisnici Connection Coprocessors ploče je mogu dobiti uz doplatu od $299 (važi do kraja godine).

# KAO U NAJBOLJIM DANIMA

**Ne tako davno ovde, na brdovitom Balkanu, za novosti iz oblasti informatike i računarstva saznavali smo iz stranih novina, i to sa više nedelja zakašnjenja, a retki strani izlagači donosili su na domaće sajmove uglavnom prospekte ili već prodatu robu. Ovogodišnji Interbiro u Zagrebu bio je bliži Evropi nego ijedan pre njega, a Jugoslavija se, zahvaljujući COMPAQ-u, našla na listi atraktivnih svetskih premijera.**

Dejan V. Veselinović

Broj izlagača je bio rekli bismo dosta uobičajeno veliki, ali samo sa više strana odnosno od više izlagača čuli da je i broj posetilaca odnosno zainteresovanih išlo tako bio neobično veliki. Naizgled, gužva je bila kao one iz najboljih dana uvođenja računara u naše živote, kada su svi dolazili da se dive tom novom čudu.

Koliko smo imali prilike da vidimo tokom jednodnevne posete, izuzev gužve, ništa nije kao u ih minulih dana. Kupaca je zaista bilo, ali su se ponašali savim drugačije no u ih pionirskih dana, veoma mnogo su pitali, dosta diskutovali, pregovarali oko cena i raspitivali se za performanse, obilazili i druge ponuđače, opet pregovarali, ali kako nam rekoše, i kupovali.

### JEDNA SVETSKA...

Stigli smo u Zagreb pola sata pre početka svečane promocije u organizaciji firme COMPAQ koja nastavlja svoj ovogodišnji gotovo neverovatni niz premijera. Kako je nas i druge obavestio gospodin Želimir Ilić, jedan od COMPAQ-ovih evropskih direktora, u momentu kada su nam u Zagrebu predstavljali nove proizvode, ista stvar se događala na trideset mesta u svetu. COMPAQ je predstavio najnoviju verziju svoga zaista izuzetnog modela LTE, koji je sada nešto malo teži od starijih verzija (za oko 600 grama), ali umesto tvrdih diskova od 20 i 40 MB sadrži diskove od 30 i 60 MB, umesto maksimalno 3 MB RAM-a sada prima 10 MB RAM-a i umesto procesora 8086 odnosno 80286 na 9.54 odnosno 12 MHz, sada koristi procesor 80386SX ili P9 na ravno 20 MHz i to sa 4 KB keš memorije. Prema podacima firme COMPAQ, ova količina obezbeđuje rad bez stanja čekanja čak 93% vremena rada.

Ovaj mali računar je time postao najveći mali računar na svetu, bar koliko je nama poznato, jer ne znamo ni za jedan drugi proizvod koji se sa njim može meriti. I dalje je ostao mali, mada nešto malo teži, i dalje je oko formata lista A4, ali sa daleko najjačim procesorom u svojoj klasi i sa osetno poboljšanom grafikom, koja umesto CGA sada koristi VGA standarde. Ovaj računar nas je oduševio i u staroj verziji, novu verziju nismo probali, a redakciji je obećana na testiranje čim u Jugoslaviju stigne prvi komercijalni primerak.

### ...I JEDNA YU PREMIJERA

Od drugih novosti istakli bismo jednu jugoslovensku premijeru opet jednog prenosivog računara, ovaj put od jednog drugog starog znanca, firme Commodore. I ovde se radi o proizvodu sasvim nalik prethodno pomenutom modelu COMPAQ – mali računar, težak jedva 3,2 kg, sa VGA grafikom, brzim tvrdim diskom (23 mS) od 20 MB, 1 MB RAM-a (proširivo do 5 MB sa dva posebna modula od po 2 MB) i procesorom Intel i80C286 na 8 ili 12.5 MHz. Kutija je isto toliko dobra kao i ona od COMPAQ-a, s tom razlikom što je mat crna (što se bar nama više dopada od bež boje), a tastatura je baš onako na pet minuta, čak veoma dobra za tako mali model. Kolege iz ljubljanske firme Konim su nam obećale ovaj inače jedini primerak trenutno u Jugoslaviji (lično ga je doneo jedan od direktora firme Commodore na Velesajam). Očekivana cena ovog računara na našem tržištu bez poreza na promet je negde oko 40.000 dinara.

Napominjemo da ova mašina sadrži i po disketnu jedinicu od 3.5 inča i 1.44 MB, što nije slučaj i sa mnogim drugim modelima u ovoj klasi.

### ZRELA HARDVERSKA PONUDA

Od ostalih prodavaca mašinske opreme tu su bili gotovo svi stari znanci od ranije, ali malo ko je imao bitno novih stvari da ponudi. Interesantan je bio štand firme Gama Electronics Zagreb, koji bi bolje bilo da se zvao Gama Electronics Yugoslavia, jer su i ostali ogranci bili prisutni. Pored već viđenog velikog televizora koji je prikazivao efekte animacije u stvarnom vremenu, bio je tu i jedan Peugeot 205 (!) sav išaran ratničkim bojama.

Štand firme MicroM/Mlakar&Co je bio daleko skromniji, a veoma teško se nalazio, znak raspoznavanja je bila velika gužva oko njega i na samom štandu, ali ni tu nismo videli ništa neuobičajeno, sem podužeg spiska ponuđenih programskih paketa koji je tako veliki da se malo ko može pohvaliti takvim brojem paketa. A ni cene nisu strašne, mada jesu veće nego one u S.A.D.



Da li zbog prezentacija, nečeg drugog ili kombinacije tog dvoga, tek štandovi prodavaca računara COMPAQ su bili blago rečeno jako posećeni. To se posebno odnosi na štand zagrebačkih firmi MikroLAB i Tera, na kojima je sedeo gotovo čitav proizvodni program firme COMPAQ, uključujući i najnovije čudo LTE 386S.

Štand firme Repro (ogranak bivše domaće a sada austrijske firme Avtotechna), sa svom Roland i Epson opremom je takođe bio u stanju potpune okupacije, da nije bilo velikog panoa, ne bismo ga ni našli. Samo ime Epson je izgleda bilo sasvim dovoljno da privuče mnoge zainteresovane za stampače ove svetske firme.

Od hardvera primetili smo na sajmu i NEC prenosnik sa kolor ekranom, ali se pokazalo da su komentari američkih kolega sasvim tačni, okanite se te boje još za neko vreme.

Zagrebačka firma Perihard, koja se posvećeno bavi ergonomskim i zdravstvenim aspektima rada sa računarima, izložila je takođe kompletnu ponudu. Neumorno su demonstrirali svoje proizvode, od više vrsta filtera za monitore, preko posebne papučice za odmor nogu, do ručice za naslanjanje lakata. Inače pohvalili su nam se sa najnovijom opremom (čiju cenu bolje da ne znate) za merenje raznih vrsta zračenja kojima je korisnik izložen. Što nas je, priznajemo, veoma obradovalo, jer je uvek zadovoljstvo videti ponekoga ko se posvećeno bavi nekim problemom, ko taj posao radi sa (još uvek) velikom dozom i entuzijazma a ne samo ekonomske računice i, konačno, nekoga ko to tako dobro i kvalifikovano radi.

I najzad, štand beogradske firme Ibis Sys koja je prikazala, koliko smo videli, ama baš celu poveću gamu u principu veoma dobrih do odličnih Mannesmann-Tally štampača se takođe izdvojio po celovitosti prezentacije. Tamo je bar četvoro ili petoro ljudi stalno pokretalo ovaj ili onaj štampač za nekog od mnogobrojnih zainteresovanih posetilaca.

### PLATI I NOSI

Toliko o hardveru. No, glavne novosti su bile na terenu softvera. Naravno, nama potrebe da naglašavamo da je čislo softverskih kuća bilo veoma malo u odnosu na ukupan broj izlagača, a među njima (bar onima koje smo za jedan dan uspeli da obiđemo) daleko najviše stvari se odigravalo na štandu zagrebačke firme Perpetuum. Ova mlada firma mladih ljudi zastupa svetski poznate firme WordPerfect Corporation i Nantucket Corporation u Jugoslaviji, a od hardvera firmu Dell Computer Corp. iz S.A.D., pored toga razvijaju i sopstvena rešenja i programe, a odnedavno se bave i razvojem na bazi programa Oracle, kao dileri za Hrvatsku. Da je ova firma počela kao čislo



Svetska premijera: COMPAQ LTE 386s/20

softverske i da se od početka bavila konsnici ma u interakciji sa programima postaje jasno čim dođete na štand

Ciklično organizovane demonstracije programa *Nantucket Clipper 5.0* su bile izuzetno dobro posećene. Na ostalim računarima su se vrteli razni programi firme *WordPerfect* sa obavezna bar tri do četiri posmatrača. U Perpetuumu nam kažu da su prodali veći broj primeraka programa no što su se nadali, mada su u rezervi imali još. U tome se ogleda i druga značajna novost – počela je masovna prodaja softvera odmah na licu mesta.

Da ne bude zabune, prodaje je bilo i ranije na sajmovima, ali su tada prodavci dolazili sa po tri ili četiri kopije, pa ko stigne, stigao je, a ostali neka čekaju. Ovo je prvi put da nam neko pokaže lager od bar po pedesetak kopija nekoliko programa, koje posetioci istina mogu kupiti i u samom gradu, ali i tu, na sajmu, odmah platiti i poneti sa sobom. Ako ćemo pravo, krajnje je vreme i bilo, ali dobro je da je konačno i to počelo.

Najzad, tu je i pitanje samih kupaca. Iz razgovora sa kolegama na raznim štandovima kao i sa nekima od kupaca, stekli smo utisak da trend sazrevanja domaćeg tržišta pokazuje sve zdrave znake eksponencijalnog rasta. Kupci se sve više respituju, sve više ih zanimaju stvari koje se pre godinu dana nisu ni spominjale, pa je to, bar za nas, sasvim sigurno najlepša vest koju možemo da saopštimo. Takvo ponašanje kupaca je dovelo i dovodi do sve većeg odziva odnosno prilagođavanja prodavaca, primera radi, pored već obaveznih ćevapčića, lepinja, hamburgera i ostalih nečistoća između štandova sa prehrambenim proizvodima videli smo i jedan štand koji je po zaista veoma povoljnim cenama prodavao BASF diskete od 360 KB (20 din./kom.) i 1.2 MB (28 din./kom.), što su gotovo minhenske cene. Tri četvrtine naše ekspedicije se odmah i po kratkom postupku snabdelo disketama, samo još jedan u nizu dokaza da je „plati i nosi" filozofija počela da se primenjuje i kod nas, umesto starih „udri ga, pa to je samo kupac" i „prvo plati a posle kad dobiješ" pristupa.

Dakle, ovogodišnji Interbiro je bio manje sajam tehničkih inovacija u oblasti računarstva a mnogo više sajam sazrevanja industrije kao celine, dakle i kupaca i prodavaca. Neki pozitivni trendovi koje smo na beogradskom sajmu samo nazirali sada su već postali sasvim jasno prepoznatljivi, a mi mislimo, verujemo i nadamo se da je to dobro – veoma dobro za sve nas.

# U VENTURI NIŠTA NOVO

Pojava nove verzije nekog „velikog" programa obično je praćena veoma kontradiktornim vestima o novostima koje donosi. Pre pojave Venturine verzije 2.0 imali smo prilike da slušamo gotovo bajke – čak i zvanična lista novosti koju je objavio Xerox obećavala je puno. Kada je verzija bila pred nama, pokazalo se da je novosti zaista puno ali da su one daleko od onoga što se očekivalo. Verziju 3.0 od početka prate još konfuznije vesti – te Xerox je definitivno prešao na *Windows* okruženje i više nema standarnog GEM interfejsa, te ipak će GEM biti zadržan, te biće dve verzije itd. Kada se *Ventura* konačno pojavila, iznenađenja je zapravo bilo malo. Zaista postoje dve verzije programa, jedan sa već poznatim GEM okruženjem i druga za rad pod *Windows*-om. Za sada smo imali prilike da vidimo samo ovu prvu, pa ćemo izneti prve utiske.

Program se isporučuje na 8 3.5 inčnih disketa i organizacija datoteka na njima je drugačija – najveći broj delova programa je spakovan svima poznatim PKZIP arhiverom ali se instalacija i dalje odvija na poznat način. Nakon umetanja prve diskete i startovanja VPPREP (ako instalaciju ide sa disk jedinice B, onda VPPREP B:) i posle već poznatog pitanja o oznaci diska na koji želimo da instaliramo program, pojaviće se prva razlika. U verziju 3.0 je u stvari upakovano ono što je u verziji 2.0 bilo prodavano kao posebna opcija – profesionalna ekstenzija se sada može birati prostim odgovorom na pitanje, a isto se odnosi i na instaliranje podrške za rad pod mrežom. Iako pri pitanju o instalaciji profesionalnog dodatka stoji obaveštenje da je za njega potrebno imati EMS memoriju, ništa se u stvari nije promenilo u odnosu na verziju 2.0. Program će raditi i bez EMS memorije ali ćete sada morati dobro da pročešljate AUTOEXEC i CONFIG datoteke i izbacite sve što se izbaciti može jer program zahteva nešto više memorije nego ranije. Naravno, profesionalna ekstenzija bez EMS memorije je kao i ranije praktično neupotrebljiva.

Instalacija programa za rad pod mrežom zahteva dva koraka – jednu instalaciju na serveru i zatim pojedinačne instalacije na svakoj radnoj stanici. Od mrežnih operativnih sistema su podržani Novel, 3 Com i PC NET. Iako nismo bili u situaciji da program stvarno instaliramo pod mrežom, ipak smo izvršili server instalaciju. Nakon startovanja programa dobili smo obaveštenje da je program detektovao da mreža nije u pogonu i da će nam biti uskraćene neke opcije. Ipak, dalji rad se mogao nastaviti na uobičajen način.

Ostatak instalacije je sasvim poznat. Pregledom VENTURA direktorijuma nakon završenog posla otkrićete da nema ničeg posebnog novog – ista kolekcija fontova, isti moduli za konverziju formata slika i teksta itd. Uz program se takođe isporučuje i standardan BITSTREAM FONTWARE paket koji je potpuno identičan sa onim iz verzije 2.0.

Nakon ovako razočaravajućeg početka startovanje programa je donelo potpuno razočarenje. Pregled menija je otkrio da nema ama baš nikakve razlike između verzije 2.0 i 3.0. Zapravo, jedina razlika koju smo primetili je nova opcija u PRINT masci – Duplex Options – namenjena štampanju dvostrukih strana spremnih za povezivanje (samo na štampačima koji ovo podržavaju). *Ventura* nas je u ranijim verzijama navikla da ugrađuje opcije bez implementacije ili samo delimično implementirane. U verziji 2.0 nije naizgled ostalo „rupa" koje bi verzija 3.0 trebala da popuni pa nam jedino ostaje da se nadamo da su mnoge primećene slabosti (naročito situacije kada program jednostavno „puca" bez nekog očiglednog razloga ili pomalo „neurotično" ponašanje fusnota) korigovane. Da li je stvarno interno nešto poboljšano pokazaće se tek za koji mesec nakon što ova verzija prođe kroz svakodnevne muke. Ono što smo za sada probali je pokazalo gotovo identično ponašanje sa starom verzijom – problematične CHP datoteke koje su obarale sistem obaraju ga (doduše na malo drugačiji način) i sada, a ponašanje fusnota takođe ne pokazuje neke razlike.

Ipak, kao što smo rekli, tek nakon nekoliko meseci upotrebe moći ćemo sa sigurnošću da utvrdimo da li razlike zaista postoje. Za sada je dobra vest da je nova verzija kompatibilna što se formata TXT, CHP, STY i WID datoteka tiče tako da se lako može uporedo pratiti ponašanje.

Dakle, što se GEM verzije tiče, nema nikakvog jasnog razloga zbog čega je napravljen tako veliki skok u oznaci verzije. Verujemo zato da je ova izmena motivisana ukupnim promenama – ugradnji svih dodataka u osnovni paket i paralelnom verzijom za *Windows*. O ovoj poslednjoj kruže, kako smo već navikli, sasvim proverene vesti o potpuno novoj koncepciji koja daleko prevazilazi GEM verziju. Dok je ne dobijemo na testiranje ostaje nam samo da se, opet po običaju, nadamo da je to zaista i tačno.

*Zoran Životić*

# JOŠ 48 IGLICA

**U poslednjem broju „Računara" opisali smo dva devetoiglična matrična štampača. Ovoga puta pažnju usmeravamo na nešto moderniju tehnologiju – na dva 24-iglična matrična štampača, od kojih se jedan veoma lako nalazi na našem tržištu, a drugi morate poručiti iz inostranstva. Zajedničko im je samo to što oba potiču iz Japana i što oba proizvode svetske kuće.**

Dejan Veselinović

Egzotičan član ime je Panasonic KX-P1124. Ako čitate „Byte" ili „PC Magazine" dobro ga znate sa brojnih reklama koje se već podože objavljuju u tim časopisima. U njima se proizvođač hvali da je štampač rezultat raznih želja kupaca, a pošto se ta reklama vrti bar godinu i po dana, ovom modelu smo pristupili kao zrelom proizvodu. Drugim rečima, naša su očekivanja od ovog modela bila veoma velika.

### AKADEMSKO PITANJE

Slično, ali u čak većoj meri se odnosi i na Epson LQ 550. I ovde se radi o veoma zrelom modelu koji nastavlja tradicije prethodnog modela LQ 500, svojevremeno najjeftinijeg Epson ovog 24-igličnog štampača. Pored toga, ovaj štampač ima i dopunski zadatak da brani samo ime Epson, koje se vremenom nametnulo kao standard kompatibilnosti i etalon prema kome se svi ostali mere. Povrh svega, ovaj model (a istini za volju i Panasonic) su trebali da pokažu kako se porede sa našim starim Epson LQ 850 modelom, koji je ostao naš referentni standard za odlične (mada i proporcionalno skupe) 24-iglične štampače A4 formata.

Prvo smo probali Panasonic KX-P1124, bojeći se pre svega potpunog odsustva bilo kakvih iskustava sa proizvodima ove firme. Veoma brzo se pokazalo da je svaki strah potpuno neosnovan, bar što se štampača tiče, ali malar nas je opet strefio – naša I/O karta je naprosto ispustila dušu, naravno, u nedelju popodne (Slava joj!). Kao pravi saradnik „Računara" superiorno smo se nasmešili i iz ormana izvadili drugu sličnu kartu i nastavili sa poslom. O tempora, o mores, printer sacra lames!

### ČIME SE HVALE...

Počnimo od modela Epson jer se on hvali sa daleko manje stvari od Panasonic-a. Ono što Epson nudi je već tradicionalni kvalitet, pouzdanost i u našem slučaju, obzirom da smo štampač dobili od zastupnika (Avtotehna/Repro Ljubljana), ugrađeni naši znaci. Niko (ni zastupnik, ni proizvođač) se ne hvali posebno nežnim mehanizmom za papir, mada u priručniku jasno navode kakve se sve vrste papira mogu koristiti. I to je sve.

Panasonic se u svim svojim reklamama za ovaj štampač bog zna kako hvali čitavim spiskom zahteva kupaca kojima su izašli u susret. Ovaj štampač ima jednu relativno retku osobinu i jednu veoma retku osobinu. Relativno retka osobina je podešavanje na čisto elektronski način, u kom smislu na prednjoj komandnoj tabli postoji niz membranskih tastera i čitav ratni plan kako da ih upotrebite. Ima ljudi kojima se ovo sviđa, a pažljivi čitaoci će se setiti da smo prethodno kritikovali štampače firme Mannesmann-Tally zbog sličnog pristupa. Pa ipak, Panasonic se nije baš potpuno izgubio u elektronici kao M-T, ali nam se ovakav aranžman i dalje ne dopada. Ovo nije samo žalo što se sve radi elektronski, nego pre svega žalo što se ovaj štampač ne može ni porediti sa modelima Mannesmann po kvalitetu mehanike. Nesrećan izbor.

Epson je sa svoje strane nastavio da razvija svoj sistem DIP preklopnika. U starijim generacijama su te preklopnike, obično dve grupe od po 8 komada, iz dubina kućišta prineli zadnjoj ivici na kojoj su onda napravili prosek kako biste im lakši pristupili spolja. Ovaj put su stvar doveli do logičnog ishoda time što su preklopnike postavili odmah iznad komandne table na prednjoj strani štampača i pokrili ih malom dekorativnom klapnom. Ovo rešenje već počinje dosta tipično za Epson proizvode, sa njim smo se sretali i kod njihovih računara, kako stonih tako i prenosivih. Nama se ovakvo rešenje najviše dopada: jeste staromodno, ali sa aspekta prosečnog (dakle tehnički ne preterano zainteresovanog) korisnika, tvrdimo da je ovo najbazbolnija rešenje. Dakle, jesna prednost Epson-u.

Druga i još ređa osobina Panasonic štampača je njegova sposobnost da uvlači pojedinačne listove A4 papira i sa prednje strane, naime, na dnu kućišta se nalazi jedna klapna koja se može ili držati otvorenom (veoma malo utiče na povećanje dubine), ili se može u celosti odvojiti od kućišta i ostaviti u ormanu. Sistem zaista radi i mora se priznati, veoma je pogodan za sve situacije u kojima se korisnik nalazi blizu štampača.

Oba proizvođača se već poslovično hvale naizgled velikim brzinama rada, koje kada se prevedu u stvarni život, u kome ima prebacivanja u novi red, različitih veličina slova i kvaliteta štampe, padnu na daleko niže nivoe od deklarisanih. Koliko, pogledajte i sami na priloženoj tabeli.

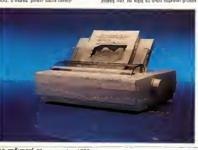
### ... A ŠTA PREĆUTKUJU

Epson ne spominje prilično nežnu mehaniku kojom je opremljen model LQ 550. Primera radi, bočni zupčanici bi se mogli lako oštetiti i malo tvrđim noktom, a to teško da je pohvalno (istina, oni samo vode papir, a vuče ga veliki valjak, ali ipak).

Sa svoje strane, Panasonic se ne hvali bukom koju proizvodi štampač, a sigurno se ne može pohvaliti mehanizmom za uvlačenje papira sa prednje strane. Po pravilu nam je list papira bio bar malo izgužvan, iako smo koristili 80-gramski papir, dakle vrednost bližu gornjem nego donjem delu skale. Samo jednom smo probali da upotrebimo laganiji papir, ali kada smo bili primorani da ga u delovima vadimo iz štampača, odustali smo od daljih pokušaja.

Epson koristi standardnu kasetu sa trakom koja se proteže gotovo celom širinom štampača, dok Panasonic koristi manju kasetu koja se kreće zajedno sa glavom. Jasno je da je prvi pristup bolji, uprkos činjenici da prema podacima proizvođača traka Panasonic-a traje 50% duže od Epson-ove. O problemima nabavke ne-Epson format traka da i ne govorimo.

Oba štampača, naravno, imaju priručnike. Ne bismo mogli nazvati nijedan od njih lošim, ali jasnu prednost dajemo onome koji se dobija uz Epson LQ-550. Profesionalnije je urađen, sadržajniji je i bogami iscrpniji. Međutim, sa zadovoljstvom konstatujemo da se opšti nivo priručnika japanskih proizvoda veoma osetno popravio, ako je suditi po ova dva priručnika: nema više nebuloznih konstrukcija na nekom



son LQ-550 - č, ć, ž, š, đ, Č, Ć, Ž, Š, Đ, 17 cpi, bold,

pson LQ-550 - č, ć, ž, š, đ, Č, Ć, Ž, Š, Đ, 15 cpi

Epson LQ-550 - č, ć, ž, š, đ, Č, Ć, Ž,

e Epson LQ-550 - č, ć, ž, š, đ, Č,
lined,

je Epson LQ-550 - č, ć, ž
cpi, bold, underlined,

je Epson LQ-550
Č, Ć, Ž, Š, Đ,
erlined, *italic*

KX-P1124 - ü, å, é, ä, ö, Ü, Å, É, Ä, Ö, 17 cpi, bold,

c KX-P1124 - ü, å, é, ä, ö, Ü, Å, É, Ä, Ö, 15 cpi,

sonic KX-P1124 - ü, å, é, ä, ö, Ü, Å,
*italic*

asonic KX-P1124 - ü, å, é, ä, ö,
rlined,

Panasonic KX-P1124 - ü, å
, 7 cpi, bold, underlined

e Panasonic KX-P1
ö, Ü, Å, É, Ä,
underlined, *ital*

kandži engleskom, količina čisto korisničkih informacija je veća i tome slično.

## EPSON, MIRNE DUŠE

Ne znamo koliko će reproduktivna tehnika u štampariji verno preslikati uzorke slova, ali posmatrajući kvalitet ispisa nema nikakve sumnje da je Panasonic još uvek daleko iza Epson-a po kvalitetu ispisa i obliku slova. Istine za volju, malo ko može da konkuriše Epson-u po tome, sem (koliko znamo) modela NEC i Toshiba.

Nemamo nikakvih zamerki sa emulacijama koje Panasonic nudi: probali smo obe i ni jedan jedini put nismo naišli na bilo kakav problem oko štampanja. Inače, kao mali kuriozitet navodimo činjenicu da Panasonic, pored uobičajenih fontova Times Roman (Courier po njima) i Helvetica, takođe serijski ugrađuje i Script i Pica, čime se emulacija modela Epson LQ 2500 zaokružuje. Pošto Epson LQ 550 nudi istu emulaciju ali nema te fontove (odnosno može ih imati posredstvom opcione kartice), proizlazi da Panasonic celovitije emulira LQ-2500 od samog Epson-a.

Što se grafike tiče, Epson je jasnije otkucao nijanse (mada je razlika marginalna), ali je Panasonic dao daleko ravnomernije slike. Naime, Epson se odavno muči sa izbegavanjem horizontalnih linija između dva prelaza glava, čak toliko dugo da se već ozbiljno pitamo muče li se oni uopšte sa time. Prema rezultatima ovog štampača, pre bi se reklo da se ni ne trude više. U slučaju brzine rada u grafičkim režimima, Epson je više nego dva puta brži od Panasonic-a, i ne samo to, već je model LQ-550 gotovo isto toliko brz koliko i kudikamo skuplji model LQ-850.

# Unitest 2 - Računari

**Panasonic KX-P1124 i Epson LQ-550**

**Novembar 1990. godine**

© TV Enterprise/Računari, 1990

| Proizvođač, model | Panasonic KX-P1124 | Epson LQ-550 |
|---|---|---|
| Dimenzije, ŠxVxD | 430x143x359 | 418x140x347 |
| Težina, neto kg. | cca. 6,5 | cca. 6,0 |
| Potrošnja struje | cca. 120 VA | cca. 100 VA |
| Glava, iglica | 24 | 24 |
| Glava, vek trajanja | 100 mil. znak. | 200 mil.znak. |
| Traka, vek trajanja | 3 mil.z.draft | 2 mil.z.LQ |
| Bafer: - standardan | 7 | 8 KB |
| - proširiv do ... | 32 KB | 32 KB |
| Papir: - A4 pojedinačno | Da | Da |
| - A4 sa doturačem | Opcija | Opcija |
| - A4 sa traktorom | Da | Da |
| - A3 sa traktorom | -- | -- |
| Vezoici: - serijski RS232C | Opcija | Opcija |
| - paralelni | Da | Da |
| Emulacije: 1. | Epson LQ-2500 | Epson generički |
| 2. | IBM Propr. XL | (vidi tekst) |
| 3. | -- | -- |
| Mogućnost rada u boji | -- | Da |
| Standardni fontovi: | | |
| - 20 znakova/inčn | Da | Da |
| - 17 znakova/inčn | Da | Da |
| - 15 znakova/inčn | Da | Da |
| - 12 znakova/inčn | Da | Da |
| - 10 znakova/inčn | Da | Da |
| - 7,5 znakova/inčn | Da | Da |
| - 6 znakova/inčn | Da | Da |
| - 5 znakova/inčn | Da | Da |
| - Proporcionalna štampa | Da | Da |
| - Kurziv/Masno/Podvučeno | Da | Da |
| - Kombinacija efekata | Da | Da |
| - Sadrži YU slova | -- | Da |
| Grafika: | | |
| - 120 x 72 tačke/inču | Da | Da |
| - 150 x 150 tačaka/inčn | Da | Da |
| - 180 x 180 tačaka/inčn | Da | Da |
| - 360 x 180 tačaka/inčn | Da | Da |
| - 360 x 240 tačaka/inčn | Da | Da |
| - 360 x 360 tačaka/inčn | -- | -- |
| Radni uslovi: | | |
| - temperatura, | 10-35 st.C | 5-35 st.C |
| - vlažnost vazduha | 30-80 % | 15-80 % |

| Proizvođač, model | Panasonic KX-P1124 | Epson LQ-550 |
|---|---|---|
| Brzina, draft, nominalna | 192 zn./sec. | 180 zn./sec. |
| Brzina, draft, stvarna | 96 zn./sec. | 92 zn./sec. |
| Brzina, LQ, nominalna | 63 zn./sec. | 60 zn./sec. |
| Brzina, LQ stvarna | 41 zn./sec. | 44 zn./sec. |
| Brzina štampanja grafike | 107,43 sec. | 43,94 sec. |

Ispis Epson-a je daleko lepši od Panasonic-ovog, ali kvalitet rada sa tim štampačima se tu ne iscrpljuje. Mehanika kod modela LQ-550, uprkos zamerci da deluje veoma nežno, u praksi je daleko preciznija od konkurenta. Zapravo, najveća mana Panasonic-a je njegova mehanika, koja je neprecizna, gruba i bučna.

Zbog svega toga, savetujamo vam da model Panasonic KX-P1124 zaobiđete u odsta velikom luku, sem možda za cenu sličnih 9-igličnih štampača, od kojih je sigurno bolji. Ovaj model još mora dugo da se razvija i poboljšava pre nego što postane bilo kakva konkurencija sličnim modelima firme Epson.

Epson LQ-550 je jedan veoma solidan 24-iglični štampač koji je nasledio uglavnom sve dobre osobine svoje velike porodice. Relativno nežna mehanika upućuje na zaključak da se radi o strogo kućnom modelu namenjenom onima kojima treba veoma kvalitetan slovni ispis i dobra do vrlo dobra grafika, ali bez kvantitativnih preterivanja. Problema sa pobudom po definiciji nema, jer se nalazi svuda u svetu pa i kod nas, a naša slova koja ugrađuje Avtotehna/Repro su praktično istovetna kao i originalna, što će reći odlično napravljena. Najzad, iako je mehanika nežna, veoma je precizna: nema nikakvih problema sa podešavanjem margina. Zbog svih tih razloga, ovaj štampač preporučujamo mirne duše.

Redakcija „Računara" se zahvaljuje firmi „Repro" na hardveru koji nam je stavljen na raspolaganje za potrebe ovoga testa. Za sva obaveštenja o „Epson" štampačima obratite se na adresu „Repro", 61107 Ljubljana, Celovška 175, tel. 061/552-341, 552-150, 554-450, fax 061/552 563.

# VRATA U PROFI SVET

**Sa starim hardverom je kao i sa starim odelom – ono se može prekrojiti da izgleda kao novo, ali se na kraju uvek dođe do zaključka da je najbolje odelo ipak novo odelo. Kontroler laserskog štampača LaserPort PS 600 je samo izuzetak koji potvrđuje pravilo. Podizanjem finoće ispisa sa 300×300 na 600×600 tačkica po inču, ovaj gotovo neugledni hardverski dodatak će preporoditi vaš tehnološki već prevaziđeni LaserJet, pretvarajući ga, za malo novca, u DTP mašinu bez premca na tržištu.**

Jova Regasek

Računari i laserski štampači u segmentu tržišta koje se naziva stono izdavaštvo poslednjih godina vrše strahovit uticaj na grafičku industriju. Nakon prvog talasa euforije visokim mogućnostima i niskom cenom nove tehnologije pripreme za štampu, danas niko više nije siguran koliko štampana reč stonim izdavaštvom dobija, a koliko gubi. Dovoljno je pogledati bleda, iskrzana slova u brojnim knjigama da bi se videlo koliko je laserska priprema i izražajnim mogućnostima za grafičko oblikovanje i kvalitetom otiska daleko ispod dostignuća savremene grafičke tehnologije. Finoća ispisa od 300×300 tačaka po inču je prećutno usvojena kao svetski standard za laserske štampače, a na tim rezolucijama jedino rasterska tehnika (izv. bit-mapirani tonovi) daje podnošljive rezultate. Konstrukcija slova u ovoj rezoluciji je, međutim, previše gruba da se njegova nesavršenost ne bi mogla uočiti na iole finijoj hartiji. Grafički jezik *PostScript*, koji ima izuzetne mogućnosti za grafičko oblikovanje stranice, daje na niskim rezolucijama veoma loše rezultate, koji nimalo ne ohrabruju stonog izdavača da zaroni u njegov tamni vilajet i savlada njegovu zatvorenost i neprilagodljivost, kao i da uloži dosta truda da bi ovladao njegovim mogućnostima.

## LASERPORT PS600

Na tržištu grafičke opreme postoji ogroman jaz između amaterskih laserskih štampača rezolucije 300×300 i profesionalnih foto-slog uređaja rezolucije 2540 i 1270 tačaka po inču. Sa izuzetkom američkog Varitypera VT600W, za čijih (kažu užasno sporih) 600×600 tačaka treba platiti preko 15 000 dolara, ne postoji ništa što bi popunilo ovaj prostor. Poslednjih meseci na tržištu se pojavilo nekoliko kontrolera koji iz postojećeg standardnog harvera izvlače finoću ispisa i do 1000×1000 tačaka po inču. Jedan među njima je i LaserPort PS600 američka firma DT Tek. Ovaj kontroler ne nudi ni najfiniji ni najbrži ispis u ovoj ponudi, ali obezbeđuje za najmanje novca (dva do tri puta manje!) briljantan otisak i, što je podjednako važno, definitivno uvodi svoje korisnike u standardni svet *PostScript*-a, koji na rezolucijama 600×600 već sasvim lepo dolazi na svoje.

Laser*Port* je hardverski kontroler laserskog štampača sa mehanizmima tipa Canon SX, čiji je najslavniji predstavnik, bez sumnje, HP LaserJet Series II. Korišćenjem pomoćnog ulaza

## OVAKO IZGLEDA

### TAJMS

| BITSTREAM 300×300 RASTERSKI FONTOVI | BITSTREAM 300×300 POSTSCRIPT | BITSTREAM 600×600 POSTSCRIPT | ULTRASCRIPT 600×600 POSTSCRIPT | LINOTYPE 2450×2450 |
|---|---|---|---|---|
| Ovo je jednostavna proba kvaliteta ispisa. Ovako izgleda *kurziv*, ovako **polucrno**, a ovako ***polucrni kurziv***. Tu su i naši specijaliteti: Žabalj i žabac, šuma i Šabac, čvarak i Čačak, Ćuprija i ćurka, Đevđelija i đavo | Ovo je jednostavna proba kvaliteta ispisa. Ovako izgleda *kurziv*, ovako **polucrno**, a ovako ***polucrni kurziv***. Tu su i naši specijaliteti: Žabalj i žabac, šuma i Šabac, čvarak i Čačak, Ćuprija i ćurka, Đevđelija i đavo. | Ovo je jednostavna proba kvaliteta ispisa. Ovako izgleda *kurziv*, ovako **polucrno**, a ovako ***polucrni kurziv***. Tu su i naši specijaliteti: Žabalj i žabac, šuma i Šabac, čvarak i Čačak, Ćuprija i ćurka, Đevđelija i đavo | Ovo je jednostavna proba kvaliteta ispisa. Ovako izgleda *kurziv*, ovako **polucrno**, a ovako ***polucrni kurziv***. Tu su i na‹i specijaliteti: Øabalj i ›abac, ‹uma i ðabac, ªvarak i «a¢ak, »uprija i "urka, ãev"elija i "avo | Ovo je jednostavna proba kvaliteta ispisa. Ovako izgleda *kurziv*, ovako **polucrno**, a ovako ***polucrni kurziv***. Tu su i naši specijaliteti: Zabalj i zabac, šuma i Šabac, čvarak i Čačak, Ćuprija i curka, Sevselija i |

### HELVETIKA

| BITSTREAM 300×300 RASTERSKI FONTOVI | BITSTREAM 300×300 POSTSCRIPT | BITSTREAM 600×600 [illegible] | ULTRASCRIPT 600×600 POSTSCRIPT | LINOTYPE 2450×2450 |
|---|---|---|---|---|
| Ovo je jednostavna proba kvaliteta ispisa. Ovako izgleda *kurziv*, ovako **polucrno**, a ovako *polucrni kurziv*. Tu su i naši specijaliteti: Žabalj i žabac, šuma i Šabac, čvarak i Čačak, Ćuprija i ćurka, Đevđelija i đavo | Ovo je jednostavna proba kvaliteta ispisa. Ovako izgleda *kurziv*, ovako **polucrno**, a ovako ***polucrni kurziv***. Tu su i naši specijaliteti: Žabalj i žabac, šuma i Šabac, čvarak i Čačak, Ćuprija i ćurka, Đevđeli a i đavo. | Ovo je jednostavna proba kvaliteta ispisa. Ovako izgleda *kurziv*, ovako **polucrno**, a ovako ***polucrni kurziv***. Tu su i naši specijaliteti: Žabalj i žabac, šuma i Šabac, čvarak i Čačak, Ćuprija i ćurka, Đevđelija i đavo | Ovo je jednostavna proba kvaliteta ispisa. Ovako izgleda *kurziv*, ovako **polucrno**, a ovako ***polucrni kurziv***. Tu su i na‹i specijaliteti: Øabalj i ›abac, ‹uma i ðabac, ªvarak i «a¢ak, »uprija i "urka, ãav"elija i "avo | Ovo je jednostavna proba kvaliteta ispisa. Ovako izgleda *kurziv*, ovako **polucrno**, a ovako ***polucrni kurziv***. Tu su i naši specijaliteti: Zabalj i zabac, šuma i Šabac, čvarak i Čačak, Ćuprija i curka, Sevselija i |

***Bitstream*** **na nekoliko načina, *UltraScript* i foto-slog. Razlike u korist rezolucije 600×600 očigledne. Iz poređenja sa foto-slogom jasno je zašto govorimo o vizuelnom utisku strane a ne i o tipografskom kvalitetu. LaserPort se vizuelnim utiskom zaista približava foto-slogu, ali je svojim kvalitetom još uvek daleko od njegovih 2450 tačaka po inču. Svi primeri su za veličinu pisma 9 punkta.**



**Na granici verovatnog. Mimo svih očekivanja i svake logike, LaserPort štampa, iz *Ventura*, pod potpuno istim uslovima, skenirane crteža znanto preciznije - levo je crtež štampan u rezoluciji 300×300, a desno u rezoluciji 600×600**

u štampač LaserPort isključuje kompletnu elektroniku i preuzima potpunu kontrolu laserskog mehanizma u štampaču sa jednim jedinim zadatkom – da poveća finoću ispisa. Kombinovanjem nekoliko tehnika LaserPort uspeva da podigne rezoluciju ispisa sa standardnih (i, kako se do nedavno pogrešno mislilo, konačnih ) 300×300 tačaka po inču na čitavih 600×600. Horizontalna rezolucija se postiže povećanjem frekvencije laserskog zraka (tzv. „dot rate"), čime se obezbeđuje ispis dvostruko većeg broja tačaka po inču, a vertikalna dinamičkim preplitanjem redova (tzv. „dinamic interleave") – ubacivanjem po jednog međureda između svakog od 300 redova u standardnoj rezoluciji. Uz sve to, kontroler na sličan način kako se to radi u štampačima HP Series III manipuliše veličinom zrnaca u toneru da bi povećao preciznost krivih i kosih linija. Rezultat je ispis izuzetne preciznosti i oštrine.

LaserPort se isporučuje na dve iznenađujuće jednostavne pločice. Jedna među njima, kontroler štampača, ugrađuje se u računar – 16-bitni slot AT tipa – a druga u pomoćni ulaz na laserskom štampaču, do kog se dolazi skidanjem metalne pločice pored serijskog ulaza. Na kontroleru se nalaze mahom PAL čipovi (programabilni logički nizovi dosta elementarnog sastava), a na kartici koja ulazi u štampač jedan čip sa osam nožica, četiri voda i četiri otpornika! Komunikacija između ova dva modula obavlja se posebnim kablom – preko njega se vrši kompletno upravljanje mehanizmom u štampaču i prenos svih podataka. Standardna veza sa štampačem, bilo serijska bilo paralelna, ne igra u ovom prenosu nikakvu ulogu. Dok je aktivan LaserPort, ovi ulazi naprosto miruju.

Fizička ugradnja LaserPorta je izuzetno jednostavna i brza. Logička ugradnja se ostvaruje odgovarajućim softverom za podešavanje sistema. Ovim softverom LaserPort se vezuje za neki od LPT portova, obično za LPT2 ili LPT3, dok LPT1 ostaje rezervisan za standardne funkcije štampača. Ovakvim konfigurisanjem laserskom štampaču se dodaljuju dvostruka uloga – preko LPT1 zadržava svoje standardne funkcije u rezoluciji 300×300, a preko LPT2 obezbeđuje poboljšani ispis u rezoluciji 600×600. Standardna i pridodata funkcija, pri tom, nimalo ne konkurišu jedna drugoj i jako se dobro podnose. Ovakvim konceptom očuvana je kompatibilnost štampača ne samo sa postojećim aplikacijama i instalacijama u okviru njih, nego i sa već završenim poslovima.

LaserPort ima isključivi zadatak da upravlja laserskim zrakom i pogonskim mehanizmom u laserskom štampaču. Da bi mogao to da radi, on očekuje binarnu sliku potpuno formirane strane u memoriji računara. DT-Tek ne isporučuje poseban softver koji obavlja ovaj zadatak, nego se oslanja na postojeće softverske emulatore *PostScript* štampača. Za sada se samo dva *PostScript* interpretera isporučuju sa drajverima za LaserPort kontroler – *Freedom of Press* i *UltraScript*, pri čemu se DT-Tek s razlogom opredelio za ovaj drugi. *UltraScript* se isporučuje u paketu sa kontrolerom, ali mu se cena posebno zaračunava i ne predstavlja obaveznu opciju. Time je DT-Tek ostavio odrešene ruke i sebi i svojim kupcima da, kada se i ako se uopšte na tržištu pojavi odlji *PostScript* interpreter, odaberu prikladnije rešenje.

## ULTRASCRIPT PC

*UltraScript* je jedan od najpoznatijih kloniranih *PostScript* interpretera na tržištu. Njegov proizvođač, firma QMS, poznata po istoimenim *PostScript* štampačima, garantuje stopostotnu kompatibilnost sa modelima Apple LaserWriter i QMS-PS 810 i svim značajnijim aplikacijama koja podržavaju *PostScript* – od *Aldus PageMaker*-a i *Microsoft Excel*-a, preko *WordPerfect*-a 5.0 i *Microsoft Word*-a, do *AutoCAD*-a, *Harvard Graphics*-a i, razume se, *Venture 2.0*. *UltraScript*, dakle, prihvata sve ono što prihvata i bilo koja druga standardna *PostScript* mašina. Uz drajver za LaserPort, koji nas u ovom času jedino interesuje, uz *UltraScript* se isporučuje čitav niz drajvera za štampače najrazličitije klase – od HP LaserJet Series II do matričnog „epsona". *UltraScript*, dakle, može da se izvršava čak i na najprimitivnijim štampačima – korisno prilikom štampanja kontrolnih kopija – ali sve ispod 600×600 ne predstavlja nimalo uzbudljivo iskustvo.

## OBRTANJE STRANE

Većina poslova u stonom izdavaštvu otiskuje se na pausu sa koga se direktno prave štamparske ploče. Obrtanjem ispisa kao u ogledalu, tako da otisak dođe u neposredan kontakt sa pločom, znatno se poboljšava kvalitet štampe. Kod rasterskih fontova ovaj posao uspešno obavljaju programi Zorana Životića iz serije HPRO. Ovi programi se, na žalost, ne mogu primeniti i u *PostScript*-u, ali on za to ima svoje sopstveno rešenje. Procedure

0 759 translate 1 -1 scale

uspešno obrću ispis, ali samo iz DOS-a, odnosno u sopstvenim *PostScript* programima. U *Venturi* ove naredbe nemaju dejstvo jer ih ona, nekom od procedura iz prologa, resetuje.

## KOLIKO KOŠTA

LaserPort PS600 kod proizvođača DT-Tek – 3031 W Pawnee, Wichita, KS 67213, (800) 727-3130, (316) 945-8600 – 899 $ bez softvera, odnosno 1 195 $ sa paketom *UltraScript PC Plus*.

Sistem radi na i286 i i386 mašinama sa najmanje 5.5 MB radne memorije i tvrdim diskom sa najmanje 4 MB slobodnog prostora. Veći kapacitet memorije poboljšava performanse, a veći prostor na tvrdom disku komfor u radu.

Neko će s pravom primetiti da navedena cena predstavlja prvi deo investicije. Drugu polovinu predstavlja cena memorije koja, za rad sa *Venturom* i *Bitstream PostScript* fontovima, ne može biti manje od 6-8 MB. *PostScript* traži puno memorije i njena nabavka se ne može izbeći bez obzira na primenjeno rešenje. Razlika je jedino što je kod drugih rešenja ona u štampaču ili na posebnoj kartici i što se može koristiti samo za štampu. Kada se memorija nalazi u računaru i kada *UltraScript* nije aktivan, možete je koristiti za šta vam drago. To jednu od najvećih mana – potpuno oslanjanje na resurse računara – pretvara u značajnu vrlinu.

*UltraScript* se logički ugrađuje u sistem na istovetan način kao i LaserPort – interpreteru se dodeljuje jedan LPT port za softverski ulaz i jedan LPT za hardverski izlaz. Svaku *PostScript* datoteku koju dobije preko određenog ulaza, *UltraScript*, nakon formiranja strane, prosleđuje hardveru na određeni izlaz. Softverski ulaz u interpreter i hardverski izlaz mogu biti vezani na isti port, na primer na LPT2. Iako se podešavanje softvera svodi na određivanje nekoliko jednostavnih elemenata – izbor štampača, izbor porta i odvajanje memorije – instalacija je predstavljala izuzetno traumatično iskustvo. Sistem je ugrađen u jednu 386 mašinu na 33 MHz sa 64 K procesorskog keša i 8 MB radne memorije kojom upravlja *QEMM 5.0*. Hardverski potencijali su daleko iznad (ubrzo će se pokazati ne baš ni toliko daleko) donje granice koju zahtevaju LaserPort i *UltraScript*. Zahvaljujući ne samo nejasnim nego i kontradiktornim podacima iz uputstva za upotrebu, sistem je odbijao poslušnost na najčudnije načine, prijavljujući apsurdne greške koje, razume se, nisu obuhvaćene zvaničnim spiskom grešaka iz uputstva.

*UltraScript* radi u tzv. zaštićenom režimu (protected mode) procesora i286/i386, koji omogućuje procesoru da razvije svoju punu snagu, sa memorijom *extended* tipa, koja se nalazi u prirodnom adresnom prostoru mikroprocesora iznad 1 MB. DOS aplikacije, po pravilu, rade u tzv. stvarnom režimu (real mode), u kome emuliraju procesor 8086, sa memorijom *expanded* tipa u adresnom prostoru u okviru 1 MB, kojom upravlja neki od EMS menadžera, na primer *QEMM 5.0*. *UltraScript* je izuzetno neumeren potrošač memorije: sam interpretar zahteva 1.5 MB (640 K + 800 K *extended*), a LaserPort za formiranje stranice A4 formata 4.2 MB, što praktično znači da sistemu treba obezbediti 5 MB *extended* memorije. Pošto *QEMM* svu raspoloživu memoriju pretvara u *expanded*, bilo je logično da mu se nadodm MEMORY=2000 dozvoli da uzme samo 2 MB. Logično, međutim, nije uvek i tačno – *UltraScript* očekuje upravo suprotno: svoju *extended* memoriju može da odvoji samo od već odvojene *expanded*. *QEMM*-u treba, dakle, dozvoliti da uzme svu memoriju, a potom *UltraScript*-u odvojiti u procesu instalacije barem 5.5 MB. To rešava samo pola problema. *UltraScript* ne ume da stavi sistemu do znanja koliko je memorije zauzeo, niti je u stanju da uoči koliko je memorije zauzela određena aplikacija. Da bi se izbegli konflikti u memoriji, upotrebu memorije treba ograničiti aplikaciji. Na primer, da bi stvar proradila sa *Venturom*, kako valja, u VPPROF.BAT datoteku treba ubaciti prekidač /E=2048, čime se *Venturi* dozvoljava da koristi

samo 2 MB *expanded* memorije. Kada se jedanput kroz nju prođe, procedura podešavanja sistema je i jednostavna i kratka. Utoliko pre ne vidimo razloga zašto se za nju nije našlo nekoliko redaka u uputstvu za upotrebu.

*UltraScript* (verzija 2.1, decembar 1989.) dolazi u nekoliko varijanti. Osnovna varijanta USPC omogućuje, kroz sistem menija, podešavanje sistema, štampanje *PostScript* datoteka i interaktivno programiranje u *PostScript*-u. Svrhe ove varijante se iscrpljuje podešavanjem sistema. Štampanje *PostScript* datoteka se jednostavnije vrši namenskim programom USPRINT, koji troši znatno manje memorije za svoj rad ili, još bolje, starom dobrom DOS naredbom COPY, a interaktivno programiranje je znatno komfornije iz *XyWrite*-a ili bilo kog drugog tekst procesora čiji se ASCII izlaz može preusmeriti na LPT2. Pored štampanja *PostScript* datoteka, koje uz znatno manji utrošak memorije može da generiše ili sam *UltraScript* ili bilo koja aplikacija, na primer *Ventura*, *UltraScript* dozvoljava štampanje iz DOS-a i, u principu, iz bilo koje DOS, GEM ili Windows aplikacije. Stepen kompatibilnosti je šarolik i, po priznanju samog QMS-a, nije uvek stoprocentan.
Verzija koju smo imali na testu ne radi, na primer, sa *Windows 3.0*, a sa ranijim verzijama ovog programa radi.

Pogonski softver *UltraScript*-a ima dve odvojena modula i oba su rezidentnog tipa. Osnovni modul se zove PCAPTURE i dug je nekih 20 K. Njegov zadatak je da prihvati sve što stiže na dodeljeni port (na primer, LPT2), prosledi to *PostScript* interpreteru, prihvati rezultate obrade i pošalje ih ili na štampač ili ih odloži u odgovarajuću datoteku. Odlaganje u datoteku („štampanje u fajl") je funkcionalno potpuno isto kao i slična funkcija u *Venturi*, a naziv datoteke i njeno mesto na disku se određuju u procesu podešavanja sistema. Drugi modul, bezimeni kod dužine nekih 50 K koji proizvodi glomazni program po imenu USLOAD, zadužen je za štampanje iz aplikacije. Kada je ovaj kod prisutan u memoriji, PCAPTURE automatski preusmerava izlaz iz *UltraScript*-a na štampač. Startovanjem oba programa LPT2 postaje *PostScript* port sa visokokvalitetnim spisom, ne utičući pritom ni na jednu standardnu funkciju štampača koje se ostvaruju preko LPT1.

## IZ VENTURE I DOS-a

Svi testovi *UltraScript*-e obavljeni su sa programom *Ventura 2.0*. Pored standardnog LaserJet Series II štampača, u procesu inicijalizacije *Venture* odabran je kao drugi štampač PostScript i vezan za LPT2. Vrsta i kvalitet štampe se biraju jednostavnim izborom štampača (LaserJet ili PostScript), porta (LPT1 ili LPT2) i WID tabele (OUTPUT.WID ili POSTSCPT.WID) iz ponude u meniju Options/Printer Info, pri čemu se se jedne na drugu vrstu može u toku jedne štampe prelaziti proizvoljan broj puta. Iako i sama ima enormne memorijske apetite, *Ventura 2.0* lepo podnosi gubitak od 72 K, koliko joj uzima rezidentni deo *UltraScript*-e, ako joj se odvoji barem 250 K EMS memorije. U DOS-u je izbor još jednostavniji – COPY *datoteka.ime* LPT1 ili COPY *datoteka.ime* LPT2. Naravno, pri tom treba voditi računa šta se na koji port šalje.

*PostScript* zahteva izuzetno brze mašine. Jedan od razloga njegove slabe popularnosti, ne ni najvažniji ni jedini, je i taj što je zahtevao više nego što je postojeći hardver mogao da mu pruži. S pojavom brzih 386 mašina i kontrolera za podizanje rezolucije laserskih štampača, moglo bi da se kaže da je hardver pola ko sustigao softver. *UltraScript* je u *PostScript* familiji posebna priča. On se, za razliku od laserskih štampača i nekih kontrolera koji imaju svoj poseban hardver, u potpunosti oslanja na resurse računara i koristi ih do njihovih krajnjih granica. U radu će se brzo pokazati da preporučeni minimum od 5.5 MB radne memorije nije dovoljan i da nije sve jedno kakav je tvrdi disk i naročito koliko na njemu ima slobodnog prostora. *UltraScript* intenzivno koristi tvrdi disk, pa je njegova brzina izuzetno važna, a preporučeni minimum od 4 MB slobodnog prostora predstavlja praznu priču. Bez obzira na odabrani režim rada – datoteka ili štampač – *UltraScript* uvek kompletan posao najpre štampa u datoteku us.tmp, da bi tek potom njenim čitanjem gradio stranicu po stranicu knjige. *PostScript* je izuzetno opširan programski jezik i proizvodi veoma glomazne datoteke, naročito ako u dokumentu ima i slika. Dok je *UltraScript* aktivan, čitav računar je zamrznut, a rezultati u pogledu brzine ipak nisu nimalo spektakularni. Brzinu je, zapravo, teško i meriti, jer ona presudno zavisi od grafičkog izgleda strane. Stoga bi možda najtačnija ocena bila: izvanredan kvalitet bez značajnih gubitaka na brzini štampe.

*UltraScript* se isporučuje sa 25, a *UltraScript PC Plus* sa 47 *PostScript* fontova. Uz obavezne (i zajedničke) helvetiku, tajms, kurijer i simbol, uz Plus verziju se isporučuje i nekoliko knjiških pisama, kao i font sa takozvanim „dingbatima" – simbolima za grafička ukrašavanja teksta (zvezdice, telefoni, uokruženi brojevi). U *PostScript*-u broj fontova nije isto što i broj pisama – u poseban font se računaju i stilovi u okviru jednog pisma (normal, kurziv, polucrno, polucrni kurziv), pa se stvaran broj pisama dobija deljenjem broja fontova sa četiri, naravno pošto se prethodno odbiju pisma koja imaju manje od četiri stila. Tako u Plus verziji ima 14 pisama u 47 stilova. *UltraScript* fontovi su kompatibilni sa Adobe fontovima, ali uprkos tome interpreter, po priznanju samog QMS-a, ne prihvata Adobe(!). QMS biblioteka fontova je, naravno, najjednostavniji način da korisnik poveća fond svojih pisama, pa, shodno takvom uverenju, QMS nije pripremio nikakvu softversku alatku za ugradnju fontova drugih proizvođača.

## REZIDENTNI FONTOVI

Ni uz *Venturu*, koja je opremljena drajverima za *PostScript*, ni uz sam *UltraScript*, ne isporučuje se softverska alatka za upisivanje fontova u štampač (download). Takvu alatku Adobe isporučuje uz svoje fontove za „mekintoš", ali od tog saznanja nemamo puno koristi. Bez takve alatke rad sa dopunskim fontovima praktično nije moguć. Font se može poslati *PostScript*-u jednostavno DOS naredbom COPY *font.ime* LPT2 /b. Interpreter se neće pobuniti (dobar znak!), ali ni shvatiti da je font u njega upisan (nikakva korist!). Da bi to shvatio, treba mu izričito reći „ovaj font sačuvaj u memoriji". Sve svoje procedure, a fontovi su takođe procedure, *PostScript* čuva u svom sistemskom rečniku. Sledeći niz naredbi na početku fonta:

serverdict begin 0 exitserver

rezerviše u sistemskom rečniku prostor za font koji sledi i upisuje ga u njega.

Ovaj skup naredbi važi za sve procedure, bez obzira na njihov broj i posao koji obavljaju, koje treba trajno upisati u *PostScript*. U takav skup spada i tzv. prolog (PRE2.PS) – zbir svih procedura koje *Ventura* koristi i koje inače šalje pre početka svake strane. (Ako se ovaj prolog trajno upiše, uštedeće se 12 K, koliko je dug, po svakoj štampanoj strani – 100 strana = 1.2 MB na tvrdom disku i dosta uštede na vremenu.)

Pri tom *UltraScript* prihvata samo jednu rezidentnu datoteku, a svaki font se nalazi u posebnoj datoteci. Pošto broj procedura i veličina datoteke nisu bitni, ako je prethodno za *UltraScript* odvojeno dovoljno memorije, rešenje se nameće samo po sebi: dovoljno je sve fontove skupiti u jednu datoteku (merge) po imenu, na primer, tajms.ps, iza prvog komentara

% PostScript Outline Font (c) Bitstream

dodati odgovarajući skup naredbi

serverdict begin 0 exitserver

i datoteku upisati u *UltraScript* nakon njegovog aktiviranja programima USLOAD i PCAPTURE, prostim

COPY tajms.ps LPT2

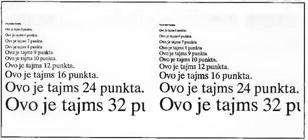
Od tog časa fontovi postaju dostupni svemu što se šalje preko LPT2, bez obzira da li to bio DOS ili *Ventura*. Pri tom za *Venturu* sadržaj datoteke POSTSCPT.CNF više nije bitan. Iz nje treba izbaciti sve fontove, a opcijom Options/Add/Remove fonts sve upisane fontove proglasiti rezidentnim.

Tehnika je izuzetno pogodna za *Bitstream* fontove jer su zapisani u najčistijem ASCII-ju. Na osnovu ovog opisa nije teško zamisliti jednostavan program koji, nakon što pošalje navedeni programski red, objedinjuje fontove i procedure po unapred pripremljenom spisku, datom, na primer, u datoteci DOWN.CFG. Ovo objedinjavanje svodi se za ukidanje koda za kraj (1AH) na krajevima svih datoteka osim u poslednjoj.

## IZ TAMNOG VILAJETA

Fontovi bez sumnje predstavljaju najosetljiviji deo priče o *PostScript*-u. Ne samo zato što su oni najvažnije ogledalo kvaliteta jednog *PostScript* sistema. Osnovna rešenja u vezi sa fontovima (tzv. tvrdi Adobe koncept) najneposrednije utiču na primenljivost *PostScript*-a na određenim grafičkim poslovima i u određenim – jezicima. Sva privlačnost *PostScript*-a pada u vodu, a njegova snaga kao *de facto* industrijskog standarda za grafički opis strane gubi svaki značaj u trenutku kada otkrijete da na njemu ne možete da napišete tako obične i trivijalne reči kao što su „ćurka", „čvarak" ili „đubre". Za razliku od nezgrapnih ali potpuno prilagodljivih rasterskih (bit-mapiranih) fontova koji se mogu prepravljati do mile volje, slova u *PostScript*-u su dobro zaštićena i svojim kriptografskim matematičkim opisom i odsustvom gotovo bilo kakve dokumentacije, i ne mogu se menjati. Biblioteka slova je relativno siromašna, znatno siromašnija od *Bitstream*-a, i u njoj se, od pet slova koji su nam od interesa, nalaze samo slova „ž" i „š". Kombinovanje karakterističnih slova YU latinice od slova i akcenata koji postoje u fondu nije tako jednostavan zadatak i u ovom času ne znamo kako bi se mogao izvesti.

Rešenje ovog problema dolazi sa sasvim neočekivane strane. Uz rasterske fontove (datoteke sa nastavkom BCO) Bitstream omogućuje i raspakivanje *PostScript* fontova (datoteke sa nastavkom BEZ), a u njegovoj bogatoj i raznovrsnoj biblioteci slova nalazi se sve što nam je za YU latinicu (i mnoge druge latinice) potrebno. Program FONTWARE, koji se besplatno isporučuje uz *Venturu*, ima i svoju *PostScript* opciju, koja raspakuje regularne i *PostScript* i ekranske fontove. U kodnoj tabeli (CSD) za *PostScript* kodovi slova nisu ispre




**Tajms u veličinama o 3 do 32 Rasterski *Bitstream* fontovi u rezoluciji 300×300 i *PostScript* *Bitstream* fontovi u rezoluciji 600×600**

mešlani kao u kodnoj tabeli za rasterske fontova. Kodna tabela YU CSD je pripremljena pomoću programa CSD EXE *Zorana Životića*, koga je za ovu priliku sam autor neznatno modifikovao. Raspakivanje jednog pisma traje izuzetno kratko ispod jednog minula a na disku ne zauzima više od 50 K po stilu. Tipografski kvalitet ovih fontova je u odnosu na fontove ugrađene u *UltraScript* posebna priča koja ne mora da bude nepovoljna za *Bitstream*. Pitanje ćirilice još uvek ostaje otvoreno. Fontovi su dati u čistom ASCII obliku kao *PostScript* programi, a za njihovo korišćenje u *Venturi* dovoljno je povezati WID tabelu sa WID tabelom za *PostScript*. Barem je u prvim probama izgledalo da je dovoljno samo to.

U radu sa rasterskim štampačima *Ventura* na početku svake strane šalje sve fontove koji se koriste na toj strani. Koncepcija *PostScript*-a je drugačija – on ne tretira ravnopravno ugrađene i spoljašnje fontove - pa je drugačiji i *Venturin* način komunikacije sa njim. Umesto da pošalje sve fontove na početku strane, *Ventura* šalje spoljašnji font iznova svaki put kada dođe do bilo kakve promene u stilu ili pismu. Ako, makar to bila i jedna jedina reč, ima deset promena, *Ventura* će pre svake promene, dakle deset puta, poslati kompletan odgovarajući font, što znači da će (za jednu reč!) napraviti us.lmp datoteku od pola megabajta! Pošto su ovakve promene u svakom tekstu prilično česte, očigledno je da bi za jednu publikaciju na tvrdom disku bio potreban prostor reda gigabajta. Da stvar bude neprijatnija, sve ovo jako zadržava *UltraScript* – navodi ga da stalno prazni keš za slova i generiše ih uvek iz početka. Prva proba sa desetak stilova na strani bile su gotovo komatozne. *UltraScript*-u je trebalo desetak minuta da odštampa stranu uz tako intenzivno habanje tvrdog diska da je izgledalo da se stvar neće dobro završiti. Za istu stranu sa *UltraScript* fontovima (ali bez naših slova) dovoljno je dvadesetak sekundi.

Problem je odmah bio jasan, ali ne i rešenje. Očigledno je da se fontovi na neki način moraju ugraditi u sistemski rečnik interpretera (daunloadovati i proglasiti rezidentnim). Iako se ova tehnika strogo preporučuje na više mesta u uputstvu za *Venturu*, nigde nema ni slova o alatki (poput HPDOWN za rasterske fontove), pa čak ni tehnici koja bi se za to mogla koristiti. Dobija se utisak da *Ventura* to potpuno automatski radi na osnovu sadržaja datoteke *POSTSCPT.CNF* u koju se upisuju svi fontovi zajedno sa putokazima na disku, koje treba upisati u štampač, ali to nije tačno. O tome nismo našli nikakvih podataka ni u tehničkom uputstvu za *UltraScript*, ni u *PostScript* literaturi koja nam je tih dana bila dostupna. Da ne dužimo, rešenje je dato u prilogu „Rezidentni fontovi". Rezidentni fontovi naravno zauzimaju određeni prostor u memoriji računara, koji se mora dodati na preporučeni minimum od 5,5 MB. Jedan font zauzima oko 50 K, što znači da se po jednom pismu mora rezervisati oko 250 K.

## PRAVA STVAR

Priču o mukotrpnom osvajanju *UltraScripta* smo mogli i da preskočimo i objavimo samo gotova rešenja. Sa osvajanjem softvera je kao i sa zuboboljom – jako boli dok traje, ali se zaboravi čim prestane. Hteli smo da do kraja ispitamo njegovu prilagodljivost i da u izvođenju konačne ocene pođemo od najvećeg nedostatka baterije LaserPort/*PostScript*: glavni problemi u eksploataciji sistema, pri čemu većina među njima nije karakteristična samo za naše podneblje, ne mogu se rešiti na korisničkom već zahtevaju rešenja na programerskom nivou. U ovom tekstu i u prilozima uz tekst opisana su osnovna rešenja koja su dovoljna za pokretanje i konforno korišćenje sistema. Rešenja se odnose na kombinaciju *QEMM 5.0/Ventura 2.0/UltraScript*, dakle na jednu aplikaciju pod GEM-om. DOS radi takođe lepo. U nekom drugom softverskom okruženju, na primer u *Windows* aplikacijama, mogu doći do izražaja neki drugi problemi koji će takođe zahtevati programerski pristup. *UltraScript* je prilično sebičan u odvajanju memorije i dosta je osetljiv na prisustvo drugih programa. Bitno je, među tim, da rezultat – besprekoran otisak koji opštim utiskom ostavlja strana mnogo je bliži profesionalnom foto-slogu nego amaterskom LaserJet-u – opravdava svaki trud.

U bateriji LaserPort PS 600/*UltraScript* softvar je loš partner hardveru. Preosetljivost na greške, čak i najtrivijalnija omaška u radu, se praktično ispoljava blokiranjem sistema ili otkazivanjem poslušnosti samog *UltraScript*-a do resetovanja računara. *UltraScript* intenzivno koristi tvrdi disk. Ako ga u tom poslu prekine bilo šta nepredviđeno - nestanak struje ili blokada računara - posledice mogu da budu katastrofalne: razorenu FAT tabelu neće iskrpiti nijedan disk doktor! U pratećoj datoteci READ.ME se stidljivo upozorava na ovaj problem, ali njegove neprijatnosti postajete svesni tek kada vas zadesi nevolja. Rešenje za *UltraScript* treba odvojiti posebnu particiju na disku (njemu je za velike publikacije i onako potreban enorman prostor) i na njoj držati samo sistemski softver.

LaserPort nije jedini kontroler te vrste na tržištu. Slične uređaje je do sada izbacilo nekoliko proizvođača. Neki među njima imaju superiornije performanse – veću rezoluciju i/ili veću brzinu – ali daleko prednjače i svojim cenama reda 3000 i više dolara. LaserPort obezbeđuje skoro bismo mogli da kažemo profesionalni kvalitet za najmanje novca: investicija od kojih 1200 dolara će naprosto preporoditi tehnološki već pomalo zastareli LaserJet i odložiti nabavku potpuno nove opreme barem za dve godine. Rezolucija od 600×600 tačkica je, iako se iz Canon SX mehanizma može izvući i mnogo više, izabrana optimalno. Skok u finoći ispisa je dovoljno veliki da se štampana strana vizuelnim utiskom približi foto-slogu, ali i dovoljno grub da dozvoli korišćenje najklasičnijih materijala – papira za foto kopiranje i pausa – i standardnog tonera. Kod kontrolera sa većom rezolucijom, kao što je LaserMaster 800×600, finoća ispisa dolazi do izražaja tek na specijalnom papiru i sa specijalnim sitnozrnastim tonerom. Imajući u vidu kvalitet i cenu koju za njega treba platiti, potpunu *PostScript* kompatibilnost i pristupačnost repro materijala, LaserPort smatramo izuzetno privlačnim za naše tržište.

Redakcija „Računara" se zahvaljuje firmi "COMPEX" na hardveru i softveru koji nam je stavljen na raspolaganje za potrebe ovoga testa. Za sva obaveštenja o „Laser Port PS600" korisnicima obratite se na američku adresu "COMPEX", Cambridge, MA USA, tel. (617) 354 5045, fax. (617) 864 9518 ili na beogradski telefon (011) 432-495

# KAD KOMPJUETRI ČITAJU

**Unos podataka predstavlja, bez sumnje, najmukotrpniju operaciju u korišćenju računara. Računari su izvanredni u mnogim poslovima, ali im čitanje baš ne polazi za rukom. Kada jedan OCR sistem prepozna 90% teksta to je izuzetan rezultat u domenu veštačke inteligencije, ali krajnje slab za praktičnu primenu. Najnovija generacija OCR softvera se nalazi na pragu savršenstva.**

Dejan Ignjatić

Naziv OCR (Optical Character Recognition) je kratica koja znači optičko prepoznavanje karaktera, gde „karakter" podrazumeva slovni znak. Pitanje je zašto treba prepoznavati karaktere kad oni izgledaju veoma prepoznatljivo na ekranu računara bez obzira da li su ukucani preko tastature ili su uneseni preko nekog skenera. Za poznavaoca je odgovor jednostavan: skeneri proizvode sliku, a računaru su potrebne reči, a do njih može doći jedino ako ume da ih – pročita.

## APLIKATIVNI TROUGAO

Ovakvo objašnjenje je jasno i za nedovoljno upućene, a predstavlja i objašnjenje zašto su do skoro OCR sistemi bili neoptimalni, jer se problemi veštačke inteligencije rešavaju dosta teško i na mnogo jačim konfiguracijama nego što je to PC.

Zato su mnogi potencijalni korisnici olako ignorisali izuzetne rezultate postignute silnim trudom u oblasti prepoznavanja teksta. Kada OCR sistem prepozna preko 90% teksta, to je izuzetan rezultat u domenu veštačke inteligencije, ali ni iz daleka dovoljan za praktičnu primenu. Objektivna je procena da su za komercijalnu primenu pogodni samo OCR sistemi kod kojih je verovatnoća prepoznavanja teksta veća od 99 %, a za profesionalnu primenu i od 99.9%.

OCR sistem predstavlja komplet koji sadrži računar, skener i OCR softver. Svaki od ovih elemenata utiče na ocenu o pogodnosti primene takvog sistema. Računar utiče na tu odluku svojom brzinom, pa proizvodjači softvera deklarišu brzinu rada softvera u odnosu na brzinu računara. Ipak, korisnik uglavnom računa na već postojeći računar, pa pozitivne karakteristike traži od softvera i skenera, iako većina profesionalnog OCR softvera zahteva isključivo snažne i skupe konfiguracije.

Dakle: kakav računar, kakav skener i kakav softver?

Prema našem saznanju samo dva profi OCR paketa mogu uspešno da rade na AT konfiguraciji zahvaljujući pre svega optimalnom algoritmu za prepoznavanje ili dodatnom hardveru. Izbor skenera je veliki. Profesionalnih isto tako. Većina profesionalnih OCR softvera podržava mnoge skenere, ali testovi su radjeni na HP ScanJet Plus i na Panasonic FX RS506U koji su dokazali profesionalni kvalitet. To ipak ne znači da su Ricoh, Mikrotek ili drugi nepovoljniji, ali je za takav sud potreban poseban test i posebna analiza. Za dodatne informaciju preporučujemo časopis MC br 8/90 iz augusta ove godine. U njemu je objavljena i slična analiza OCR softvera sa razlikom u izboru prikazanog OCR softvera i ocenama.

## DA LI, KADA I KOJI?

Odgovor je DA. MOŽE ODMAH, ali koji...? Radi se o tome da su sada proizvodjači OCR softvera novim verzijama dostigli nivo profesionalne aplikacije i da se bez sumnje neki od njih sa uspehom mogu primeniti.



OmniPage: Dovoljno efikasno za profesionalnu primenu

Mi smo izabrali, analizirali i testirali tri paketa koji mogu konkurisati za komercijalnu ili profesionalnu aplikaciju – *Omnipage 2.1*, *ReadRight International* i *Recognite Plus 1.1* – mada su se pojavili i novi za koje kod nas još ne postoje reference.

Od američkih programa treba pomenuti *WordScan Plus* od Calera Recognition Systems sa cenom od 995 $ i mogućnošu prepoznavanja teksta sa matričnog printera i fax-formata, kao i *Image-in* od Image In Inc Co. koji je deklarisan kao integrisani paket za OCR za konverziju rasterskog u vektorski format i za grafičku obradu, sa cenom od 795 $.

U Nemačkoj se prodaje *OPTOPUS* od 11.000 DM, koji zahteva *Concurent DOS*, minimum PC/386 sa 4 MB RAM i dodatnu kartu sa specijalnim 32-bitnim procesorom sa još 2MB RAM-a, pa ipak sa brzinom od samo 25-33 cps (znakova u sekundi). Tu je i *READSTAR 6* od Inovatic-a, koji mnogo obećava, ali sa cenom od 6970 DM. On radi sa novim „topološkim" algoritmom i to uspešno na PC/AT uz veliku deklarisanu brzinu od 180 cps.

Možda ćemo pogrešiti što ne navodimo i neke druge OCR pakete, a specijalno zainteresovane upućujemo (ako nismo cinični) da ih potraže u studiji Frost & Sullivan-a „Scanning and Optical Recognition Equipment market in Europe" uz cenu od studije od 3300 $.

## OMNIPAGE 2.1

Tekst na naslovnoj strani OmniPage paketa glasi „Latinski OMNI znači SVE:

1. Bilo koji font na bilo kakvoj stranici u skoro svim tajl-forma tima kompjutera.
2. Najmoderniji softver za prepoznavanje stranica teksta za korisnike „mekintoša" i MS DOS-a.
3. Omogućuje skeniranje svih stranica sa tekstom i grafikom."

Da, jedan je od boljih profesionalnih OCR paketa, ali ne radi na konfiguracijama slabijim od PC/386 sa najmanje 4 MB RAM-a i najmanje 8 MB slobodnih na hard disku.

Za instalaciju na PC/286 neophodna je specijalna dodatna hardverska kartica, sa koprocesorom i keš memorijom od 640 KB. Radi isključivo pod *Windows*-om. Tačnost prepoznavanja je oko 99.9 % (ali zavisi od tipa i kvaliteta teksta, kao što je slučaj i kod drugih OCR softvera). Savršeno razdvaja tekst od slike i kolonu od kolone.

Raspoznaje specijalizovane znakove „latinskih" jezika (danski, holandski, irski, švedski, italijanski, engleski... ukupno deset) ali ne i YU karaktere, kao ni slova ostalih jezika istočne Evrope.

U tekstu mogu biti mešani fontovi i tipovi pisama – proporcionalni i neproporcionalni – kao i kvaliteti otiska, uključujući i otisak sa matričnog štampača.

*OmniPage 2.1* ima najveći raspon veličine slova – od 6 do 72 tačaka – tako može da čita i krupne naslove iz novina, što nije slučaj kod drugih OCR softvera, mada drugi imaju i finiju rezoluciju (do 4 tačke, na primer *Read Star 6*). Ima veoma precizan „prozor" za izdvajanje dela teksta, ali ne i za isključenje dela teksta, što mu je mana.

*OmniPage 2.1* prepoznaje i razlučuje „ligature" – povezana slova u tekstu, mada nekad greši. Iako radi na brzim konfiguracijama, brzina prepoznavanje mu nije zavidna – deklarisana je na 40-115 cps (karaktera u sekundi), mada testovi pokazuju da je najveća brzina 34 cps. Prepoznaje i TTIF fajlove, tako da je primenjiv za prepoznavanje fax-fajlova konvertovanih u TTIF, što je odlika i drugih savremenih OCR softvera. Mana mu je što nema mogućnost učenja. Ima ugradjen stalni HELP, što olakšava rad i korišćenje programa, a i sopstveni tekst editor, tako da se korekcija grešaka vrši bez izlaska iz programa.

Podržava skenere sedam firmi: Abaton, Agfa, Canon, Desl, HP, Microtek, Ricoh i Umax, ali ne i ostale skenere iz profesionalne klase, kao što su Kyocera, Panasonic, Princeton, niti mnoge druge, verovatno zato što ne podržava rezoluciju veću od 300 dpi (tačaka po inču).

Navedene manjkavosti ipak ne smetaju oceni da *OmniPage 2.1* predstavlja jedan od boljih OCR paketa na tržištu – što se tiče tačnosti prepoznavanja i komfora u radu, ali ne i brzine! – i da se može sa sa uspehom koristiti u profesionalnoj primeni.

*OmniPage* je proizvod američke firme CEARE CORP., koja je nedavno izbacila novi OCR paket *WordScan* plus, a u Jugoslaviji za

# KAKO OPISMENITI KOMPJUTER?

**Kompjuter, najviši domet visoke tehnologije, još nije potpuno opismenjen. Kada to bude, otvoriće se neslućena područja u njegovoj primeni.**

Dejan Ignjatić

Opismeniti dete ili čoveka predstavlja dugotrajan proces učenja jednog visoko inteligentnog bića. To opismenjevanje je veoma složen proces, koji se sastoji u prepoznavanju oblika slova koje treba naučiti i, kasnije, u dodeljivanju takvom obliku određenog značenja. Porediti nivo inteligencije čoveka i mašine još nije moguće, jer broj informacija koje čovek od detinjstava primi preko svojih receptora je ogroman i neuporediv u odnosu na broj informacija koja su programeri u stanju da upišu u računar. I ne samo zbog toga!

Naučiti kompjuter da prepoznaje slova i znakova isti je proces kao i opismenjavanje čoveka, samo na mnogo nižem nivou inteligencije.

Izgleda dosta neverovatno da kompjuter nije „pismen" kada je u stanju da izuzetno dobro, brzo i efikasno obrađuje tekst u stinim tekst editorima se hiljadama tipova fontova i sve to da štampa na velikom broju različitih štampača. To je, međutim, isto kao da veštog ali nepismenog čoveka stavite da slaže slova i stranice prema prema unapred datom uzorku koji je pripremio pismen čovek, a da pri tome nepismeni nema pojma šta slaže i šta piše na stranicama koje slaže.

Opismeniti kompjuter znači omogućiti mu da prepoznaje slova koja obrađuje, a to nije jednostavan proces. Kompjuter treba da prepozna i da im dodeli pravo značenje hiljade različitih oblika slova – fontova – i stotine različitih veličina sa različitim razmacima između slova, između reči i između redova.

### Kako je počelo...

Još davne 1929. u eri mehanizacije radnih procesa nemački naučnik G Tauschek prijavio je patent pod nazivom „Upravljanje radom mašina pomoću svetlosne energije". Metoda je bazirala na foto-postupku određenih slova stvaranjem slike slova na foto materijalu i kasnije projektovanjem tih slova na mašinu koja je ta slova prepoznavala. Rezultati su bili veoma skromni i nagovestili su samo da je taj proces daleko od realne primene.

Tek nastankom snažnih kompjuterskih sistema i razvojem veštačke inteligencije tih sistema stvoreni su algoritmi koji su bili u stanju da sa više ili manje tačnosti prepoznaju značenje teksta i značenja pojedinih slova. Tada je i nastao termin „OTR" i „OCR", što je kratica za optičko prepoznavanje teksta i optičko prepoznavanje karaktera.

OTR je intenzivno razvijan za vreme drugog svetskog rata, kada je trebalo prepoznati šifrovane poruke neprijatelja, a OCR sitemi takođe uz njih, ali komercijalna primena OCR sistema postala je stvarnost tek kada je postalo moguće u kompjuter uneti sliku teksta preko optičkih čitača – skanera.

Postoje mnogi algoritmi i metode za prepoznavanje teksta i znakova, a predstavljaju domen intenzivnih istraživanja u primeni veštačke inteligencije: Markovljev algoritam, Viterbi-algoritam, Contextualni proces, Tekst segmentacija, Levenshteinova N gramska i mnoga druga tehnike koje je teško prezentirati u ovakvom članku.

Nismo ni mi u Jugoslaviji zanemarili ovaj veoma značajan domen u primeni veštačke inteligencija. Na Elektrotehničkom fakultetu univerziteta u Ljubjani razrađivan je OCR sistem za prepoznavanje slova pisana rukom. Projekat su vodili prof dr L Gyrgyek i dr N Pavešić koristeći topološki metod sintektičkog prepoznavanja uzoraka, koji se primenjuje i u najsavremenijim OCR sistemima uz razne modifikacije i optimalizacije.

### I kako će se završiti?

Topološki metod se zasniva na prepoznavanju karakterističnih delova pojedinih slova, koje se sastoji od više linija (kosih, vertikalnih, horizontalnih), od više krivina (levih, desnih, konkavnih i konveksnih) i od više debljina. Topološki metod je primenljiv samo na brzim računarima sa velikom memorijom.

U ranijim komercijelnim OCR sistemima korišćen je i metod matričnog preklapanja (matrix matching), na kojem se zasnivao i prvi prasistem iz 1929, ali je tačnost prepoznavanja bila veoma mala, tako da je sada potpuno prevaziđen.

Danas komercijalni i profesionalni OCR sistemi koriste razne varijante topološkog metoda za prepoznavanje karaktera, i to neki veoma uspešno. Tako se procenat tačnosti za dobro naučene fontove približava 100% tačnosti prepoznavanja. Pri tome je brzina prepoznavanja čak na najobičnijem PC/AT računaru do 200 karaktera u skundi. To znači da se jedna strana teksta može prepoznati i prebaciti u tzv. ASCII fajl za dalju obradu u tekst procesorima za oko 10 s, što daleko prevazilazi mogućnosti i najboljih daktilografa. To je ono čemu se teži i što će OCR sisteme uvrstiti u nezamenjive alatke u primeni kompjutera.

OCR sistemi će omogućiti ono što se doskora nije moglo očekivati: slepi će slušati tekst iz dnevnih novina, pisane poruke će izvršavati razne komande u sistemima pod kontrolom kompjutera.

Autoru ovog teksta je poveren zadatak da omogući slepima da čitaju knjige i dnevna novine, što je bila davnašnja težnja humanih ljudi, ali ostvariva tek sada uz primenu savremenih OCR sistema i sistema za sintezu ljudskog glasa. Kaže se, budućnost je počela (bar u domenu visoke tehnologije).

### Kako izabrati OCR sitem?

OCR sistemi se danas koriste najčešće u izdavaštvu, u obradi teksta sa gotovih dokumenata, u stvaranju baza tekstova raznih namena, u distribuciji dokumenata, u automatskom popunjavanju formulara i bankovnih dokumenata, a intencije i perspektiva idu mnogo dalje.

Izabrati pravilno OCR sistem znači izabrati optimalni OCR softver, a uz njega optimalni skaner i računar za određenu namenu. OCR softver koji se može svrstati u profesionalnu kategoriju mora da zadovolji sledeće:

- mora biti OMNIFONT, što znači da je unapred naučen da prepozna sve standardne fontove, tj oblika slova i karaktera koji se koriste, kako proporcionalna tako i neproporcionalna
- mora da ima tačnost prepoznavanje preko 99.5% za optimalno naučene fontove uz dobru kopiju teksta
- mora da ima veliku brzinu prepoznavanja znatno veću od brzine rada daktilografa, što iznosi najmanje 150 znakova u sekundi,
- mora da prepozna sve standardne veličine karaktera, od najmanjih do velikih, tj od 6 pa do 20 ili više tačaka
- mora da raspoznaje tekst od slika i da prevodi samo tekst, a slike da zanemari,
- mora da ima mod za doučavanje za karaktera koja nema u generičkim fajlovima i mogućnost za izradu korisničkih fontova
- mora da ima mogućnost učitavanja više strana jednu za drugom za masovan unos teksta, kao i da podržava rad skenera sa „sheet feeder-om",
- mora da podržava većinu profesionalnih i drugih skenera, kako korisnik ne bi bio upućen na samo određeni skener
- mora da učitava formulare i tekstove u više kolona
- mora da zauzme što manje memorije i dela hard diska, kako ne bi zahtevao skupu računarsku konfiguraciju za svoj rad
- mora da je neosetljiv za manja oštećenja unesenih karaktera, kao i na ligature u neoptimalnim kopijama
- mora da ima tekst procesor za doradu teksta nakon unosa ili da radi u okruženju koje mu to omogućava, npr pod *MS Windows*-om,
- mora da generiše izlaznu datoteku u takvom obliku da je mogu prepoznati najpoznatiji programi za obradu teksta
- mora da prepoznaje i druge grafičke formate, npr TIFF, da bi mogao biti primenjen za prepoznavanje faksa

Uz ovakav spisak zahteva jedan OCR paket svakako ne može biti jeftin. Međutim, ako se optimalno odabere, OCR sistem će vrlo brzo opravdeti svoju nabavku.

sada nama ponuđača. Cena u Nemačkoj iznosi oko 2500 DM za PC/386 sa instliranih 4 MB RAM-a, a 4600 DM za PC/AT kada se isporučuje sa dodatnom karticom. Može se nabaviti u Nemačkoj (na primer kod Georg Zipfel GmbH tel 089/8546020 ili u Austriji (Linz) kod SoftCon GmbH tel 0732/271190)

### READRIGHT INTERNATIONAL

Za *ReadRight* njegovi autori kažu: „Prvi low cost/high performance topološki OCR sistem. *RadRight* nudi veću tačnost, veću fleksibilnost i jednostavnost rukovanja od bilo kog OCR sistema. On je superioran i u odnosu na „matrix matching" OCR sistema i u odnosu na druga topološka sistema."

Da je bolji od drugih to kažu autori, a šta

kaže analiza?

To je takodje softver „omnifont" tipa koji se može koristiti i na PC/AT i jačim konfiguracijama PC a, a i na mekintošu

Prepoznaje azbuke 9 zemalja zapadne Evrope a nema ni mogućnost naknadnog učenja, slično kao kod *OmniPage*-a

Za ugradjena jezike tačnost prepoznavanja mu je veća od 99 5% pri optimalnoj kopiji unesanoj sa skenera Otporan je na greške i prekide u konturama znakova, a takodje prepoznaje tekstove pisane na matričnom štampaču u draft modu! Nije radjen kao *Windows* aplikacija ali radi pod *Windows*-om Brzina prepoznavanja iznosi 110 cps na PC/AT/16MHz, a 150 cps na PC/386 Pri smanjenju rezolucije skenera sa 300 DPI na 200 DPI brzina mu je veća za 25-35 procenata, ali tačnost opada

Veličina slova mu nije jača strana – prepoznaje veličina od samo 6-15 tačaka (*OmniPage* 8-72), tako da nije pogodan za unošenja velikih slova iz tekstova što pretstavlja znatno ograničenje u primeni

Razdvaja uspešno tekst od slika ali ima problema sa kolonama i višestukom vertikalnom linijom sa leve strane teksta iza koja „ne vidi" tekst Taj problem se prevazilazi upotrebom „prozora" kojima se uklanjaju ometajuće vertikala(?!)

I porad navedanih nedostataka *ReadRight* predstavlja dobar OCR softvar za masovan unos sitnog teksta obzirom na brzinu i cenu ali prvanstvano za zapadno tržište sa ograničenjem za naše tržište zbog nedostatka YU seta Može se na Zapadu nabaviti uz dosta nisku cenu a u Jugoslaviji ga nudi Hermes Ljubljana uz cenu od 1000 $ (uz HP ScanJet Plus Scanner sa cenom od 2870 $) a uz dolare idu i carina i uvozni troškovi Hermes nudi kažu i mnogo bolji *TrueScan* (uz dodatak specijalne kartice) za koji nemaju tehničkih podataka Do sada nisu prodali ni jedan primerak verovatno zbog cena od 3553 $

Hermes se u Beogradu može dobiti na telefon 011/544-299 a u Ljubljani na 061/443-278

## RECOGNITA PLUS 1.1

Proizvodjač je SZKI Recognita Corp iz susedne Madjarske, koja za svoj OCR softver kaže „To je OCR softver broj jedan na evropskom OCR tržištu Brz, jednostavan samoučeći, svestran, bez rivala i fleksibilan medju OCR"

Analize i testovi, izgleda izgleda potvrđuju visoko mišljenje proizvođača o sopstvenom proizvodu Prihvaćen je na calom zapadnom tržištu veoma dobro uključujući i SAD Predstavlja takodje softver „omnifont" tipa koji prepoznaje izmešana fontova i tipova pisama u jednom tekstu

Koristi specijalni algoritam sličan topološkom ali na bazi zatvoranih kontura što je doprinelo velikoj brzini i tačnosti prepoznavanja, ali uz visoku cenu – ne prepoznaje draft otisak matričnih štampača

U NLQ modu i u svim drugim vrstama teksta radi sa velikom tačnošću, većom nego kod drugih OCR softvera Za dobro „naučeni" font sa dobre kopije tačnost prepoznavanja je preko 99.9 % Ovaku tačnost je osvojio tek u novoj verziji i to za serije posle 14 9 1990 g U toj seriji *Recognita Plus 1 1* je za svoje susede iz Jugoslavije u internacionalnoj verziji paketa pridodala YUASCII set a i setove za sve svoje istočne susede tako da predstavlja jedini OCR softver sa YU setom i specijalnim akcentnim fontovima

Za rad je dovoljan XT a na PC/368 ima zavidnu brzinu prepoznavanja od 180-250 cps što zavisi od kvaliteta kopije teksta Na brzom



PC/AT radi sa brzinama do 200 cps

Savršeno razdvaja tekst od slika i prepoznaje i sortira tabele, ali bez mreže tabele Podržava skoro sve skenere do 400 DPI (27 proizvodjača sa više od 66 tipova skenera) Ima veliki raspon prepoznavanja veličina slova i to od 6-24 tačaka (OmniPage 8-72!) Radi pod DOS-om i *Windows*-om i u BATCH modu, a podržava i dodatnu EMS memoriju

Prepoznaje veći broj setova karaktera od svih OCR paketa koji se mogu naći na tržištu što mu daje veliku prednost a može se generisati i korisnički set karaktera ili editovati postojeći

Funkcija „učenja" značajno povecava tačnost prepoznavanja za tekstove sa sistematskim greškama ili oštecenim slovima Ima mogućnost razlučivanja legatura na do 3 povezana karaktera Pod *Windows*-om ima svoj editor prepoznatog teksta a pod DOS om nema Ima mogućnost učitavanja „landscape" i „portret" formata lista sa scenera kao i rada sa TTIF fajlovima a dozvoljava i da se postavi više prozora za izbor ili eliminisanje teksta za prepoznavanje

*Recognita Plus* ima „statistiku" rada pri prepoznavanju sa podacima o procentu tačnosti brzine i vremena prepoznavanja što drugi nemaju u toj meri (*OmniPage* ne daje brzinu i vreme?!) Kad se sve sabere, ocena za *Recognitu* je vrlo visoka i ova verzija se može preporučiti kao optimalno rešenje za profesionalnu aplikaciju Može se nabaviti na celom zapadnom tržištu sa priručnikom na jeziku zemlje distributera, a isporučuje se sa HARDLOK om za Centronics U Nemačkoj ga nudi Georg Cipiel GmbH po ceni od 2600 DM (isto kao za *OmniPage*) a u Jugoslaviji nudi i zastupa LTS iz Zemuna tel 011/190 572 po ceni od 21000 Din (uz proli Panasonic Scanner FX RS506U sa cenom od 32000 Din.)

## UMESTO ZAKLJUČKA

U priloženom uporednom „trijagramu" može se videti samo odnos cene tačnosti i brzine rada OCR softvera, a ne i sve što je dobro ili loše u njemu

Ali i to je dovoljno da se oceni da u ovom trenutku *Recognita Plus 1 1* vodi trku sa znatnom prednošću u odnosu na drugi OCR softver To i zbog toga što na cenu ne utiče samo cena softvera U nju ulazi i cena koju korisnik mora da plati za kompletan sistem, a to zavisi od zahteva softvera kao i cena vremena koje je potrebno za njegovu primenu u aplikaciji – korekciju i doradu u tekst editoru nakon prepoznavanja

# SNAGA V6 MOTORA

**PC Tools je jedan od paketa koji je u proteklih godinu dana doživio više promjena od bilo kog programa na DOS tržištu. Iz malog uslužnog programa izrastao je kroz brojne verzije u univerzalnu alatku koja vas ni na jednom poslu neće ostaviti na cedilu.**

**Davor Petrić**

Prvo da kažemo da naša verzija ima datum 2. maj 1990. a puni joj je naziv *PC Tools V6c* International. Postoji verzija *PC Tools* za USA i za prodaju van USA – internacionalna verzija. Zbog propisa američkog Ministarstva obrane zabranjeno je izvoziti program koji vrši šifriranje podataka po njihovim standardima, stoga internacionalna verzija ne može kodirati podatke (to je jedina razlika paketa).

Kao i ranije, *PC Tools* ima odličan korisnički interfejs (isti kao u verziji 5.5), vrlo dobro radi sa mišem (isprobali smo različite drajvere za miša, nije bilo nikakvih problema). Osim što miša prilikom svake promjene pozicionira u donji desni dio ekrana (što zna biti iritirajuće) na rad sa mišem nemamo prigovora.

Zahtjeva 512 KB RAM-a, potpuno ugodno se osjeća na 640 KB, ne pravi pitanje ni iz grafike, ni iz diskova niti iz vrste sistema. Spomenimo odmah i to da je program namijenjen korisnicima PC sistema sa tvrdim diskom. Oni koji imaju samo disketne jedinice neće imati koristi od njega.

## ŠTO IMA NOVO?

Na prvi pogled, dizajn kutije. Sada je decentan, nije više kao „Kviskoteka". I kutija i uputstva su ugodne svjetlo krem boje. Za program iz uslužne grupe, kutija je i velika i teška. Osim tri knjige, sadrži i dva seta disketa: 6 od 360 KB i 3 od po 720 KB. Osim prve dvije (na 3.5" disketama važi: osim na prvoj) nalaze se komprimirane datoteke.

Instalaciona procedura ne zahtjeva nikakve šifre, ali program ima svoj broj koji koristite pri komunikaciji sa CPS, kada tražite pomoć i kada nadoplaćujete za novu verziju. Kompletna instalacija traje ispod 6 minuta, a tokom nje pruža vam se mogućnost da automatski instalirate sve module. Ona unosi i promjene u vašu AUTOEXEC.BAT datoteku, čuvajući staru verziju. Postoji i opcija selektivnog kopiranja datoteka na mrežni disk i ažuriranja postojeće kopije *PC Tools* na disku.

Naravno, i kasnije možete jednostavno dodavati ili isključivati instalirane programe. Iako se datoteke mogu same (ručno) raspetljati, preporučljivo je izvršiti predvidjenu automatsku instalaciju. Instalacioni program vas obavještava i koliko svaki od dijelova *PC Tools V6* treba prostora na disku. Kompletan paket traži 2865 KB, a pojedinačno, situacija je sljedeća:

| | | |
|---|---|---|
| 1 | *DOS Shell* | 1200 K |
| 2 | *Data Recovery Utilities* | 320 K |
| 3 | *Hard Disk Backup and Restore* | 475 K |
| 4 | *Desktop Organizer* | 690 K |

Uputstva su vrlo opširna. Čak bi se moglo reći i da su preopširna, pa u toj šumi podataka nije uvijek jednostavno pronaći pravi odgovor. Knjige su organizirane po logičkim cjelinama, uglavnom slijedeći organizaciju menija. Na kraju svake je dobar indeks.

U prvoj knjizi (403 stranice) se nalazi sve o *PC Shell* modulu, te o ostalim modulima: *Mirror, Rebuild, Format, Compress, Cache* i *Secure*, te *Lap Linku*. Tu se govori i o jednom od udarnih noviteta, konkurenciji Nortonovom *Disk Doktoru* i Mace-ovom *Emergency Room*, o programu imenom *DiskFix*.

Druga knjiga (411 stranica) je u potpunosti posvećena izvanrednom modulu *Desktop* (zamjenjuje olovku i papir na vašem stolu), dok je u posljednjoj, na 183 stranice opisana upotreba *PC Backup* programa.

Paket sadrži i registracionu karticu, te simpatične naljepnice za diskove koji sadrže sigurnosne kopije (*Backup*) uradjene pomoću *PC Tools V6*. Usput, ukoliko se pokušate registrirati u Americi, vjerojatno će vas prebaciti u njihovu podružnicu u Engleskoj. Takodjer, registrirani korisnici dobijaju internacionalnu verziju za *Upgrade* (iz Engleske).

## UDARNI NOVITETI

Pošto je *DiskFix* potpuna novost, počnimo sa njim. Znate za mnoge probleme koji su se u stanju pojaviti tokom rada sa diskovima: nestanak particija, oštećenje FAT tabela ili direktorija, neprepoznavanje diska, oštećenje Boot sektora i podataka u njima i mnoge slične neugodnosti s kojima naprosto moramo živjeti. Možda vam se nisu nikada desili, ali nemojte slaviti, sigurno ćete i vi doći na red!

*DiskFix* tvrdi da ispravlja velik broj tih problema. Pošto je Norton imao veliku praksu sa svojim *Disk Doktoro*-m, za pretpostaviti je da će NDD riješiti više situacija od *PC Tools DiskFix*-a, ali to nipošto ne umanjuje vrijednost možda najinteresantnije novosti u paketu *PC Tools*.

Koliko smo ga mi uspjeli istestirati, radi pouzdano, čak u ponekom detalju i bolje od NDD (npr. omogućava izbor da li želite obrisati ili snimiti izgubljene klastere, NDD ih zna samo snimiti). NDD je u prednosti što zna sam snimiti sistemske datoteke na disk ako je to potrebno, *DiskFix* očekuje da to vi uradite pomoću SYS komande.

Na žalost, glavni nedostatak oba spomenuta programa je što, ako stvari krenu nagore i disk ne bude popravljen (sasvim moguće sa svim programima te vrste), možete sjesti i plakati. Taj je problem do sada ispravno riješio samo Mace, čiji ER zna vratiti disk u prijašnje stanje.

*DiskFix* lijepo radi i sa particijama većim od 32 MB, suradjuje i sa *Disk Manager*-om, a ne smeta mu ni kada DOS nije jedini operativni sistem na disku (za razliku od Mace ER). Zaista vrlo vrijedno poboljšanje ovoga popularnog paketa koje će mnogim korisnicima omogućiti da ožive svoje diskove sa čak i vrlo ozbiljnim simptomima, a bez ikakvog znanja o njihovim uzrocima, pa i o samim diskovima.

Drugi udarni novitet je potpuna implementacija *Lap Linka* firme Traveling Software. To je najpopularniji program za prenošenje podataka izmedju dva PC (postoji i verzija za Mac) sistema, obično izmedju „velikog" i prenosnog. Dovoljno je na oba sistema pokrenuti kratke programe, spojiti ih jeftinim null modem kablom i diskovi (i diskete) onog drugog sistema se pojavljuju na vašem ekranu kao da se nalaze unutar kompjutora sa kojim radite.

*PC Shell* je potpuno funkcionalan, sa ogradom da radi kao da je spojen na mrežu. To znači da ne možete npr. formatirati disk satelitskog sistema, ali sva kopiranja datoteka rade kao metak i to vrlo brzo. Inicijalna brzina prenosa je ujedno i najveća: 115200 bauda.

Kao i ranije, *PC Tools* radi i na mreži, poštujući pri tome pravila lijepog ponašanja na mreži. Podrazumijeva se da za svaku radnu stanicu treba kupiti posebnu kopiju, ili kontaktirati Central Point Software za mrežnu licencu.

Korisnom se može pokazati i postavljanje šifara na korisničke nivoe. Postoje tri nivoa, svaki može biti zaštićen posebnom šifrom. Svaki od nivoa ima na raspolaganju funkcije koje odgovaraju korisniku sa znanjem toga nivoa. Jasno vam je da neke mogućnosti ovoga paketa mogu biti vrlo opasne po vaše podatke ako se pogrešno koriste.

Zato tino odredite koliko znate sebe, program i potencijalne probleme, i u priličnoj ste mjeri zaštićeni od slučajnih destrukcija. Ako više osoba koristi sistem, tada svatko može znati samo šifru za njegov nivo, što povećava sigurnost podataka toga sistema.

## KAKO VAM DRAGO

Sam raspored menija se nije promijenio. Unekoliko je promijenjena organizacija njihove sadržaja, uglavnom na bolje. Lagani kozmetički zahvat u debljinu okvira prozora (sada su tanji) je doprinio većoj preglednosti ekrana i sadržaja prozora.

Vrlo nam se je dopala mogućnost odredjivanja sadržaja posljednja dva reda ekrana. Ako želite, možete imati nešto veće prozore, a samo posljednji red ekrana popunjen informacijama o funkcijskim tasterima. Druga mogućnost je da iznad spiska funkcijskih tastera, odn. u predzadnjem redu, imate stalno prikazane i skraćenice (to je ono što je u verziji 5.5 bilo na dnu ekrana).

Tu je locirana i jedna vrlo bitna novost: umjesto skraćenica, u predzadnjem redu, moguće je dobiti i DOS komandu liniju. Time je uklonjen ozbiljan nedostatak verzije 5.5. Pošto se prozori (stablo direktorija – Tree List i njihovi sadržaji File List) mogu ukloniti sa ekrana, možete raditi kao da ste pod DOS-om, bez *Shell* programa.

Tada je od *PC Shell*-a vidljiva jedino posljednja linija na ekranu, odn. ona sa opisima funkcijskih tastera. Pritiskom na ALT, u prvom redu ekrana se pojavljuje i standardan osnovni meni *PC Shell*-a. Uz DOS komandnu liniju do-

dalo je i memorisanje posljednjih 16 komandi tamo izdanih. Vrlo dobro, ali ipak ne kao u *Take Charge! 1.30* koji to ima odlično riješeno.

Po nama najbolji detalj u vezi instrukcija u ta posljednja dva reda ekrana je što se funkcijski tasteri (osim 1, 3 i 10) mogu predefinirati kako vam drago. Ponekad bi poželjeli i fleksibilniju mogućnost definiranja bilo koje instrukcije, ali i izbor instrukcija samo iz svih podmenija je dobar.

Pošto su funkcije 5 do 8 ionako duplirane (nalaze se na spisku skraćenica u predzadnjem redu) mi smo na njihovo mjesto odmah ubacili funkcije za mapiranje memorije (MMap), za promjenu sortiranja datoteka tokom prikaza na ekranu (Disply), rad sa direktorijima (DMaint) te očitavanje direktorija (TreeRd). Probajte i vi.

Pozivanjem pojedinih opcija u menijima posljednji red ekrana se mijenja i pokazuje odgovarajuće namjene funkcijskih tipki, prilagođenih izabranoj opciji. Bilo bi zgodno kada bi se i te funkcije (sadržaj posljednjeg reda) mogle definirati od strane korisnika, a ne samo one na osnovnom nivou.




## TRIDESET FORMATA

Možda najveće poboljšanje je povećanje broja viewera. Verzija 5.5 je znala posmatrati samo *Lotus*, *dBASE* i ASCII datoteke bez ulaženja u program koji ih je kreirao. Verzija V6 podržava preko 30 formata datoteka. Tu se nalaze sve popularne baze i tekst procesori. Prisutni su i skoro svi spreadsheet formati.

Nevjerojatno ali istinito, nema viewera za *Quattro Pro* datoteke – nastavak WQ1 ?! Osim svega toga moguće je gledati i datoteke komprimirane sa svim glavnim kompresorima datoteka, a čak i PCX grafičke datoteke.

Pri tome je odlično što nije potrebno nikakvo definiranje. Jednostavno zatražite pogled na datoteku, *PC Shell* će ju sam identificirati. Željeni sadržaj se na ekranu pojavljuje trenutno.

Vezano uz gledanje datoteka, spomenimo poboljšanju organizaciju prozora. Kada želite neku datoteku pogledati sa View otvarate prozor (nešto manje) od polovine ekrana (okomito ili vodoravno, ovisno kako ste definirali).

U gornjoj polovini se još uvijek nalazi spisak datoteka u tekućem direktoriju, tako da jednostavno možete mijenjati datoteke čiji sadržaj želite vidjeti. Da bi vam prozor sa pogledom pokrio čitavu površinu, dovoljno je pritisnuti tabulator i pozvati Zoom.

Kao što vidite, sada se između prozora, i kada ih je više od dva na ekranu, prelazi jednostavnim pritiskom na TAB taster (naravno miš je najelegantniji). Postoji i način da odmah popunite cijeli ekran sa pogledom na željenu datoteku.

Kada je pogled zaokupio samo pola ekrana, možete se pritiscima na TAB pozicionirati u bilo koji od tri otvorena prozora (stablo, spisak datoteka i pogled) i skrolirati taj prozor. Korisno kada želite npr. pogledati veći broj datoteka koje nisu poredane ili čak nisu u istom direktoriju.

## BOLJE NEGO PRE

Funkcije za rad sa datotekama su kao i u prošloj verziji, što znači skoro odlične. Neke boljke su ostale. Npr. uspoređivanje datoteka će srušiti sistem ako se jedna od uspoređivanih datoteka nalazi na npr. *Disk Manager* particiji većoj od 32 MB (barem kod nas). Hex editor datoteka i dalje ostaje dostatan samo za najjednostavnije poslove, nipošto ne može biti konkurencija Nortonovom *NU* ili Maceovom *MUSE*.

Lociranje datoteka je preživjelo velike izmjene. Prvo što vam se ponudi su grupe. To znači da dobijete spisak programa koji u Applications meniju imaju pridružene ekstenzije kako bi se mogli lansirati i „startanjem" datoteke vaših podataka kreirane tim programom, npr. *Quattro Pro* nastavak WQ1, ili *FoxPro* nastavak DBF. Osim grupa, postoji i nedefinirana prazna grupa, kada želite neku specifičnu pretragu.

Osim tih gotovih grupa moguće je definirati i druge, pridružujući im nazive za identifikaciju i potrebne nastavke. Kada zatražite lociranje neke datoteke izborom neke od definiranih grupa odmah suzujete potragu samo na datoteke sa npr. ekstenzijom WQ1. Dodatan komfor je i što u sljedećem koraku možete zadati i niz znakova iz same datoteke koji vas zanima.

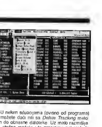
Time je učinjen veliki korak ka umanjivanju glavne prednosti *Magellan*-a, jer sada i *PC Tools* može odlično locirati datoteke kombinirajući to sa potragom za ključnom riječi. Locirane datoteke se pregledavaju pomoću viewera (osim ako se radi o *Quattro Pro*).

## POVRATAK U ŽIVOT

*PC Tools V6* ostaje program koji ima najbolje mogućnosti vraćanja izbrisanih datoteka u život. Ta je mogućnost povezana sa modulom *Mirror*, a sastoji se u instaliranju malog rezidentnog programa (8 KB) koji vodi računa o spremanju svih podataka o izbrisanim datotekama.

Koristite li DOS 3.3 ili veći, i imate dosta slobodnog prostora na disku, dosta starih grešaka će biti moguće ispraviti sistemom *Delete Tracking*. Koliko je datoteka fragmentirana nema veze. Jedino što vam mrsi konce je kada se nova datoteka prepiše preko obrisane.

Osim te, na raspolaganju vam je i standardna metoda vraćanja izbrisanih datoteka koja ne može riješiti slučajeve kada su datoteke fragmentirane. Mogućnost ručnog vraćanja je dobro koncipirana, ali smatramo da premali dio klastera vidiiv u jednom trenu na ekranu jako otežava takav rad.



U nekim situacijama (ovisno od programa) ne možete doći niti sa *Delete Tracking* metodom do obrisane datoteke. Uz malo razmišljanja, obično možete i te programe natjerati da rade na pravi način (npr. ne snimati datoteku na disk zamjenjujući – Replace staru, već radeći Backup stare).

Mali prigovor je da, kao i u verziji 5.5, nije moguće izabrati više datoteka odjednom prilikom vraćanja sa *Delete Tracking* metodom, već je potrebno vraćati datoteku jednu po jednu. Ipak, ovo je daleko najbolji način vraćanja datoteka obrisanih u toku rada (između dva dizanja sistema).

## KOMANDNI CENTAR

Tokom automatske instalacije, *PC Tools V6* će pretražiti vaš disk i u meni za aplikacije dodati poznate programe koje pronađe. Upisati će i potrebne nastavke, i spremni ste za rad. Doduše, *WordPerfect 5.1* nije prepoznao (5.0 je). Ako želite, jednostavno će biti dodati nove aplikacije.

Za razliku od prethodne verzije, sada vam je omogućen ugodan rad tokom definiranja aplikacija. Možete im čak i mijenjati raspored bez ikakvih trikova.

Kao i ranije, programu se mogu prenositi i potrebni parametri i ime datoteke koju treba učitati nakon startanja. Dobra je novost i mogućnost prenošenja pritisaka na tastere programu nakon njegova startanja. Recimo da nakon svakog startanja nekog programa morate pritisnuti ENTER / pa F5 da bi mogli početi raditi kako želite.

Dovoljno je to upisati u opis aplikacije i stvar je riješena (ne zaboravite nakon svih promjena snimiti ih na disk sa Save Configuration). Moguće je definirati i vrijeme koje je potrebno sačekati prije sljedećeg pritiska. To je korisno jer neke aplikacije isprazne bafer tastature prije prihvaćanja odgovora kako ne bi došlo do prihvaćanja pogrešnog odgovora.

Tu se krije i greška koja nije velika, ali nas nervira jer na nju nabasamo 100 puta dnevno. Ako ste sa nekom aplikacijom povezali njen nastavak (npr. za *Quattro Pro* to je WQ1) i sa željom da startanjem datoteke sa vašim podacima automatski startate i program koji ju je kreirao, nećete biti u stanju da startate samo program iz Aplications menija.

Ako se kurzor nalazi na (ne izabranoj) datoteci sa nastavkom odgovarajućim toj aplikaciji, *PC Shell* će bez pitanja učitati i tu datoteku (htjeli vi to ili ne), a ako datoteka pod kurzorom ima pogrešan nastavak za aplikaciju koju startujete, inzistirati će da potvrdite učitavanje te datoteke ili da ga zabranite. Najbolji je štos što tu poruku uopće ne možemo naći u uputstvu. U verziji 5.5 stvari nisu tako stajale.

Najveći napredak aplikacioni meni ispoljava kad je *PC Shell* instaliran kao rezidentan (mi

ga koristimo instaliranog normalno, zadnji red u AUTOEXEC.BAT nam je PCSHELL). Tada možete startati bilo koji program iz aplikacionog menija (ne i u verziji 5.5), a da bi stvari bile još ljepše, kada pozovete *PC Shell* u taj program, iz njegovog aplikacionog menija možete startati koji god program želite.

Ovo možete uraditi samo jednom (ne možete opet pozvati *PC Shell* pa startati treći program...), ali može biti vrlo korisno.

## NEVOLJE S DISKOM

*Mirror* i njegov partner *Rebuild* i dalje služe za oživljavanje pogrešno (nenamjerno ili zlonamjerno) formatiranog diska, a kada se koristi PC *Format* za formatiranje disketa, tada je moguće oživjeti i sve datoteke na formatiranoj disketi. Kao što znate, DOS *Format* prepisuje cijelu disketu novim sadržajem tokom formatiranje, uništivši nepovratno njen prethodni sadržaj.

Nije moguće dobiti *Delete Tracking* bez izvršavanja *Mirror*-a. To vas košta neko vrijeme pri dizanju sistema, zauzima stotinjak KB diska, a mnogima Mirror (odn. *Rebuild*) neće nikada zatrebati. Koliko ste puta nenamjerno formatirali tvrdi disk? Mi nikada.

Što se tiče *PC Cache* modula, na našem je sistemu postigao vrlo loše rezultate u usporedbi sa *Smartdrive 2.10* i *MCache*, ali možda tako neće biti i na vašem sistemu. Radi u svim vrstama memorije.

Modul *PC Secure* možete koristiti za vrhunsko kodiranje vaših povjerljivih podataka ako imate USA verziju, a ako imate internacionalnu verziju, koristan je kao solidan kompresor datoteka. Binarne datoteke skraćuje za oko 30%, datoteke tekst procesora i za više od 50%, čak i do 60%, dok će datoteke baze podataka ili spreadsheet programa biti stisnuta (najčešće) i na još manje.

*PC Backup* je vrlo dobar program za pravljenje zaštitnih kopija. Rad je jednostavan jer koristi isti korisnički interfejs kao i ostatak paketa. Program vrlo točno predviđa koliko će mu disketa biti potrebno za tekući backup. Snimanje na diskete je vrlo brzo ako koristite DMA način.

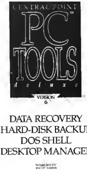
Kompresija je dobra, čak i nešto bolja od najboljeg Backup programa – *Fastback Plus*. Stvarnih 40.239.756 bajtova u 1.649 datoteka (fizički zauzima 44.544.000) sa našega glavnog DOS diska, *PC Backup* sprema na 22 diskete od 1.2 MB. Nemojte pokušavati raditi zaštitne kopije na disketama od 360 KB formatiranima kao 1.2 MB. To je moguće, ali nije nimalo sigurno. To su ipak kopije koje trebaju biti vaše osiguranje!

Kada se radi o DOS uređajima (npr. disk ili traka) rad se bitno uspon (za razliku od *Fastback Plus*). Osim što radi sa disketama i ostalim DOS medijima, program podržava i neke jedinice trake koje su zasnovane na QIC-40 i QIC-80) standardu.

Na žalost, naš strimer COREtape Light, iako radi po QIC-80, za *PC Backup* je ostao stranac u noći, pa program nismo mogli provjeriti i sa našom jedinicom trake.

Pouzdanost kopija je dobra, ali prije ozbiljnog rada, obavezno podesite DMA brzine vašega sistema (ne zaboravite Setup kompjutora), pa tek kada uspoređujući kopiju sa originalima ustanovite da nema niti jedne greške, možete početi koristiti *PC Backup* (i bilo koji drugi) kao ozbiljno sredstvo.

U slučaju oštećenja diskete, može se restaurirati sav preostali dio backupa. Diskete nisu formatirane po DOS standardu, imaju jedan




sektor više. Zato nemojte brijati sa editorima sektora po tim disketama. Sam *PC Backup* će uraditi najviše što se može kada je potrebno spašavanje sa oštećenih backup disketa.

Diskete se snimaju do posljednjeg bajta. Za razliku od verzije 5.5, sada je moguće i dodavanje novog backupa na stari. Postoji više načina, urađeni su kako treba. Datoteke je moguće selektivno uključivati i isključivati iz snimanja, mada su prostori za definiciju premali. Različite konfiguracije je moguće snimiti na disk pod različitim imenima kako bi se upotreba što više olakšala.

## UMESTO BELEŽNICE

Što se tiče *Desktop*-a, reći ćemo samo da je, po nama, najbolji program u svojoj grupi, mjereno i prema samostalnim programima tog tipa. Instaliran kao rezidentan, zauzima samo 40 KB memorije. Moguće je otvoriti do 15 *Desktop* aplikacija istovremeno, prozore u kojima rade aplikacije možete prilagođavati (i veličine i pozicije) svojim željama.

Ima bilježnicu (maksimalno 64 KB po jednoj datoteci) sa proizvoljnim brojem datoteka, outline, jednostavnu bazu podataka kompatibilnu sa dBASE III.

Financijski, naučni i jednostavan kalkulator su kao i u verziji 5.5, ali je zato programerski kalkulator postao pravi HP kompatibilan. Jedino što je kalkulatore nezgodno koristiti bez miša zbog velikog broja funkcija.

Telekomunikacije su odlične za jedan *Desktop* program. Imate program za modem zajedno sa mogućnošću skriptiranja, a moguće je slati i primati faksove.

Makro editor je vrlo dobar, a ako ste radili sa nekim od poznatijih samostalnih programa, sve će vam biti jasno i logično. Osim direktnog pisanja komandi, moguće je i snimanje pritisaka na tastere.

Vjerojatno najkorisniji dio *Desktop*-a je rokovnik. On će se brinuti da uvijek budete na vrijeme obaviješteni o obavezama. Moguće je postavljati alarme u okviru dana u određeni sat, jer je rokovnik organiziran po satima. *To Do* lista je općeniti podsjetnik (korisno za rođendane ili godišnjice, registraciju auta i sl.). *To do* lista se bazira na datumima.

Odlično je rješenje gdje se vidi i kalendar za tekući mjesec, dio *To Do* liste i tekući rokovnik odjednom na ekranu. Takav način prikazivanja omogućava odličnu preglednost.

*Desktop* je toliko interesantan da ga smatramo (a i ostale PIM – Personal Information Manager (menadžeri ličnih informacija) interesantnom temom za samostalan članak.

## SVAKAKO KUPITI

Prigovore smo uglavnom spomenuli. Voljeli bi vidjeti i zatamnjivač ekrana (kao npr. u *DesqView* ili *Take Čarge*!). On vam nakon nekoliko (zadatih) minuta neaktivnosti tastature ili miša isključi ekran, štedeći tako fosfor na zaslonu.

Bilo bi dobro i da su moduli integrirani u jedan. To se odnosi na *Mirror* Delete Tracking‹D› (8 KB), rezidentan *PC Shell* (10 KB) i *Desktop* (40 KB). Tada bi umjesto 60 KB zauzimali samo 30–40 KB), mada ni sadašnja potrošnja memorije nije velika.

Također bi voljeli da se *Desktop* učitava u manjim komadićima u memoriju, jer on, pozvan iz nekog programa, mora prvo snimiti na disk skoro 300 KB. To traje, ovisno o brzini vašega diska, nekoliko sekundi, ali bi se moglo eliminirati učitavanjem samo potrebnih dijelova i postepenim oslobađanjem memorije, ili možda primjenom odličnog Borlandovog VROOMM koncepta.

Poneka greškica u paketu *PC Tools V6* se nađe, ali sve u svemu, program je vrlo korektan. To se posebno odnosi na njegov rezidentan rad, jer nam još nije stvorio probleme niti u jednoj prilici.

Za pohvaliti je služba podrške korisnicima. Mi smo kontaktirali sa engleskom centralom. Promptno odgovaraju na vaše upite, rješavajući tako sve vaše (eventualne) probleme. Po našim iskustvima, bez dileme se nalaze među najboljim službama podrške koje smo vidjeli (uz američki Borland i kanadski Corel). Ima i firmi gdje na naša pisma ili faksove nisu odgovorili mjesecima (da ne kažemo nikada).

Nova verzija je i nešto skuplja od prethodnih. *PC Tools V6* ima preporučenu cijenu u Americi 150 USA, dok je kupovina nove verzije za registrirane korisnike samo 30 USD. Takvom cijenom je izjednačen sa ostalim programima iz te grupe.

Što se tiče njegove korisnosti, mislimo da je *PC Tools V6* vjerojatno najkorisniji program koji možete instalirati na vašem PC sistemu. Razlog je jednostavan: on ne radi nikakve velike stvari (kao baza ili tekst procesor ili spreadsheet). Ali ono što radi – bezbroj malih svakodnevnih poslova – radi odlično.

**Korisna adresa**

Central Point Software Europe LTD
Newall Road Heathrow
TW6 2EX United Kingdom
Tel. 081 897 3435
Fax 081 759 786

# NAJBRŽI KEŠ NA SVETU

**Vlasnici PC računara zaista ne mogu da se požale na manjak uslužnog softvera – sa svih strana na nas „navaljuju" komercijalni, *public domain* i *shareware* keš programi u čiji je razvoj očito uloženo mnogo znanja i pažnje. Ipak, do pre dvadesetak dana smo živeli u ubedenju da problem keširanja diska nije rešen na način koji bismo nazvali zadovoljavajućim. A onda je stigao *PC-Kwik Power Pak*...**

Dejan Ristanović

Ako prosečni vlasnik PC ja može da kaže da uslužnih programa ima više nego dovoljno šta tek da kaže neko ko ima zadovoljstvo da ih prikazuje za jedan kompjuterski časopis? Poslednjih godina kroz ruke su mi prolazile najnovije verzije komercijalnih programa praćene, naravno, svom potrebnom dokumentacijom i punom podrškom proizvođača. Svaki od tih keš programa sam startovao meno isprobavao, hvalio i kritikovao. Pa ipak uvek se postavljalo pitanje koji od tih programa trajno „nase1iti" u *autoexec*. Istorija mog *autoexeca* po pitanju keš programa je veoma duga i neprijatna.

## OD NEMILA DO NEDRAGA

Keš program pre svega ima smisla počevši od AT mašina sa *Extended* memorijom – odseći deo od inače tesnih 640 kilobajta za ubrzavanje diska nije baš prihvatljiva cena za (makar i značajno) ubrzanje kompjulera. Prvi keš koji sam koristio na AT mašini sa megabajtom memorije (640+384 *Extended*) bio je SMARTDRV SYS iz paketa *Microsoft Windows*. Radio je prilično nesrećno – ubrzanje se dakako primećivalo ali se iz nepoznatih razloga dešavalo da jedan pritisnut taster bude ponovljen desetak puta što je, naprimer, jako „pogodno" kada u *Venturi* ili *WordPerfect*-u pritisnete PgDn a kompjuter ga multiplicira. Razlog za ovakvo ponašanje nikada nije lociran (premda su se slične stvari događale na raznim pločama zasnovanim na SUNTEC čipovima – pogledajte „Bajtove lične prirode") ali se rešenje pronašlo – treba *extended* memoriju pretvoriti u *expanded* (LIM) i onda postaviti keš u nju što, inače, proizvođači keš programa najčešće i savetuju. Nevolja je bilo što moj prvi drajver za pretvaranje *extended* memorije u LIM (izvesni SEMS SYS) nije bio baš remek delo softverske industrije pa se sa loše slagao sa SMARTDRV SYS računar se s vremena na vreme „mučki" blokirao što mi je ogadilo SMARTDRV – iako se u stvari radi o veoma dobrom keš programu, zapravo možda i najboljem pre *PC-Kwik Power Paka*, izbacio smo ga iz CONFIG-a i po nabavci SEMS4 SYS i SEMS5 SYS mu se nisam vraćao.

Sledeći program (koji sam najduže koristio) bio je MCACHE iz paketa *Mace Utilities*. Svaka mu čast – radio je pouzdano i nije blokirao računar, bio je kompatibilan sa svime što sam stavio na disk ali nije bio bez mane. Pre svega, performanse su mu možda bile dobre u danima kada se pojavio ali je posle bilo programa koji mnogo više ubrzavaju disk. Zamerao sam mu i što nema parametre nego svaki put kada treba da promenim dimenzije keša treba da tražim program pomoću koga ću napraviti novi MCACHE SYS a nije se ni mogao izbaciti iz memorije kada se jednom učita premda se mogao isključiti (što mu je plus u odnosu na, recimo, PC CACHE). Prelazak na 386 me je, međutim, konačno naterao da „bacim" i ovaj keš – zaključio sam da je presport!

Sledeći korak je bio neizbežni PC CACHE iz paketa *PC Tools De Luxe*. Ovaj program uglavnom deli PC svet na dve klase ekvivalencije – oni koji ga obožavaju i oni koji ga mrze... koriste ga ipak i jedni i drugi. Dobre strane su pre svega solidne performanse (da opet dodamo – solidne, ako se ne uzme u obzir *PC Kwik Power Pak*), zatim mogućnost startovanja iz *autoexeca* uz zadavanje parametara (npr. ako dokupite četiri megabajta RAM-a, PC CACHE /SIZEXP=1024K ćete lako zameniti sa PC CACHE /SIZEXP=4096K) i, naravno, podizanje kompletnog programa u *high* memoriju pomoću QEMM-a na 386 mašinama. Strahovita mana ovoga programa je što ga ne možete isključiti – PC CACHE se istina može ukloniti iz memorije ali samo ako je rezidentan u osnovnoj memoriji i ako posle njega niste učitavali ništa drugo. Pošto to praktično nikada nije slučaj, jednom startovani PC CACHE vam ostaje „zauvek" – zbog arhitekture DOS a je jasno da se ne može ukloniti iz memorije ali bi baš mogla postojati mogućnost da se njegov rad isključi! Ovako me svaki COMPRESS diska čini nervoznim – jeste da PC CACHE navodno sarađuje sa COMPRESS om (radi se najzad o programima iz istog paketa) ali bih ipak isključio keš pre kompresije, pogotovu ako prethodno nisam napravio *backup*. Ovako se stvar svodila na stalno editovanje *autoexec*-a, dodavanje REM-a ispred linije u kojoj piše PC CACHE i naravno docnije zaboravljanje da se taj REM izbriše uz čuđenje zašto odjednom računar tako sporo radi.

Da su mu samo ovo mane, sa PC CACHE-om bi se još i dalo živeti. Brojni korisnici su međutim prijavljivali tajanstvene blokade mašine (naročito AT a mada je bilo i 386-ica) u prisustvo ovog programa – nikada to nije bilo ništa što bi se moglo potvrditi i dokazati, ali je i na Sezamu bilo dosta reči o nevoljama. To mi je nekako „ubilo" poverenje u PC CACHE ali sam ga, u nedostatku boljeg, držao na računaru. Katastrofa je nastala kada je stigao *PC Tools 6.0* i kada sam instalirao njegov keš – performanse su se poboljšale ali mi se tastatura blokirala čim bih malo intenzivnije udarao u numeričke tastere (!?). Računar tada dolazi u jedno veoma zanimljivo stanje – sve radi normalno (čak se mišem može izaći iz programa ako dotični program podržava miša) ali su dirke jednostavno mrtve dok se ne pritisne crveno dugme (Alt Ctrl Del jasno ne dolazi u obzir jer ni ti tasteri ne reaguju). Sličan problem se ponovio na još jednoj 386 mašini dok je na 3-4 ostala računara sve bilo sasvim normalno što bi značilo da krivicu treba pripisati pre nekoj interakciji između te dve (različite!) matične ploče, BIOS a i keša nego samom PC CACHE u ali je to slaba uteha – ipak je lakše promeniti keš program nego matičnu ploču! Baš kada sam došao do zaključka da dobrog keš programa jednostavno nema i počeo da prekopavam gomilu „bačenog" softvera u potrazi za PC *Toolsom 5.50* stigao je *PC Kwik Power Pak*.

## BRZINA JE ZGODNA...

Ovaj program čudnog imena (kao da je pisano po Vuku) nije novost – već dve godine osvaja nagradu *PC Magazinea*, *PC Worlda* i *Software Digesta* kao najbolji keš program za PC računare na svetu. Kod nas ga na žalost nije bilo pre svega zato što je do skora bio potpuno samostalan (*PC Tools*, na primer, sadrži keš ali i gomilu drugih stvari) a košta ne baš zanemarljivih 80-tak dolara. Oduševio me je međutim već na prvi pogled – poslednjih desetak dana proveo sam tražeći mu manu i očekujući da naletim na neku „minu". Još mi nije uspelo!

*PC Kwik Power Pak*, osim keš programa SUPERPCK COM sadrži i nekoliko drugih alatki ali su sve one integrisane u jedan paket na način koji do sada nismo imali prilike da vidimo. Odlučite se na primer, da 4 megabajta vaše LIM memorije posvetite ubrzanju sistema i prema tome u *autoexec* ugradite SUPERPCK /S 4096 /A+ (/A je neobavezno i naglašava da koristite LIM memoriju). Onda vam zatrebe RAM disk. Ništa lakše – u CONFIG SYS ste već ugradili DEVICE = PCKRAMD SYS ali vam RAM disk za sada ne uzima ni bajt memorije! Čim međutim, podaci počnu da se upisuju na virtuelni disk PCKRAMD SYS će „porazgovarati" sa SUPERPCK COM i „pozajmiti" od keša tačno onoliko memorije koliko je potrebno za smeštanje datoteka – ostatak je i dalje keš! Ako na RAM disk upišete još podataka dobićete još memorije. Ukoliko neke od tih podataka izbrišete prostor se oslobađa i vraća kešu, ukoliko uradite *del h:*.** (H: je recimo, RAM disk) keš će ponovo porasti na 4 megabajta. Sve ove operacije jasno korisnik ne primećuje – sve se dešava automatski.

Ili da za primer uzmemo printer spuler – kada podaci navale na štampač, PCKSPL COM će pozajmiti memoriju od keša i u nju odlagati ono što još nije štampano. Kako se štampanje privodi kraju tako se memorija oslobađa i vraća kešu: sasvim jednostavno ali i izuzetno efektno – pravo čudo da niko od „konkurencije" ne pokušava ništa slično.

Ovaj koncepcijski trik sa „bankom" iz koje se po potrebi pozajmljuje memorija je zaista izvanredan ali nikako nije jedini adut *PC-Kwik Power Paka* – najvažnije je da je svaki od programa iz paketa urađen izvanredno i da često ostavlja konkurentske proizvode za nekoliko dužina iza sebe. Počnimo od keša i od najjednostavnijeg merenja performansi – pustite CORETEST i PC-CACHE će dati brzinu prenosa od 3037 a *PC Kwik Power Pak* od 5451 KB/sec. CORETEST, naravno, nije namenjen testiranju keš programa pa njegove rezultate treba uzimati sa velikom rezervom – ipak pokazuje o kakvim se odnosima radi. Ukoliko hoćete da vidite preciznije podatke o brzinama, pogledajte *PC Magazine* koji je, kako rekosmo

već dva puta izabrao *PC-Kwik Power Pak* za najbrži i najbolji keš program

### ...A FLEKSIBILNOST JOŠ ZGODNIJA

Brzina je zgodna stvar, ali SUPERPCK može da se pohvali i nečim što je možda i važnije – fleksibilnošću. Hoćete da ga uklonite iz memorije? SUPERPCK /U. Ne može se ukloniti iz memorije ali hoćete da ga isključite? SUPERPCK /D. Ponovo da ga aktivirate? SUPERPCK /E. Hoćete da ispraznite keš? SUPERPCK /F. Hoćete da se i sam program podigne *expanded* memoriju? SUPERPCK /EM. Hoćete da pogledate kojim sve opcijama raspolažete? SUPERPCK /? uz jedan fini štos – ako je program već instaliran, dobićete spisak opcija koje ima smisla primenjivati „na živo", dok ćete, ako program nije instaliran, dobiti spisak opcija za startovanje kojih, istini za volju, ima mali milion. Sve opcije prirodno imaju *default* vrednosti koje su veoma razumno izabrane tako da se startovanje najčešće svodi na prosto SUPERPCK /s nnnn gde je *nnnn* veličina bloka koji želite da utrošite na ubrzanje sistema. Mi smo dodali i jedno /p, koje unekoliko skraćuje ispis na ekran pri učitavanju programa – bez ove opcije PCKWIK ispisuje trenutne vrednosti svih opcija što zauzme dvadesetak redova.

Specijalitet SUPERPCK-a je keširanje upisa. Ništa novo, rekło bi se – većina modernijih *cache* programa se, uz dodavanje odgovarajuće opcije, mogu „naterati" da i upie na disk prenose kroz memoriju čime se rad računara dalje ubrzava, naročito kada se radi sa nekim većim bazama podataka koje su višestruko indeksirane. Korisnici se, međutim, retko i nerado rešavaju da aktiviraju ovu opciju – uvek postoji opasnost da nestanak struje ili „zaglavljivanje" računara uništi rezultate čak i dugotrajnog rada koji sticajem okolnosti još „stoje" u RAM-u. Kod SUPERPCK-a opasnosti kao da nema – upis se ne kešira na klasičan način, ali se svi zahtevi za upis malo „obrade" pa se optimizuje kretanje glava diska.

Silan dobitak, rekło bi se, ali u praksi mnogo znači – jedan moj programski paket koji obrađuje baze podataka je sa PC-CACHE-om radio složeno ažuriranje za 4 minuta, a sa SUPERPCK-om za svega 2.5. Čak se i bez nekog posebnog programa može videti kako ovo (ako je verovati dokumentaciji, potpuno bezbedno) keširanje upisa deluje – u nekom punijem direktorijumu otkucajte *xcopy* *.* a: i ... čudite se!

### IZNENAĐENJIMA NIKAD KRAJA

Toliko o keš programu i RAM disku – prelazimo na sledeću komponentu zvanu PCKKEY ili ubrzavač tastature (*keyboard accelerator*). Uz klasično ubrzanje tastera sa strelicama PCKKEY omogućava i nelinearno ubrzanje tj. da se kurzor u početku kreće sporije (pretpostavka da hoćete da ispravite neku reč koja je „tu blizu") a zatim, kada „shvati" da idete daleko, sve brže i brže. Ovo je u prvom trenutku malo neobično, ali se čovek dosta brzo navikne posle čega mu obični kurzor izgleda tako trapavo i nepokretno.

Sastavni deo PCKKEY-a je editor DOS komandi – pritisak na gornju ili donju strelicu u uglu ekrana (tekstualnog ili čak grafičkog, uz podršku CGA, EGA, VGA i Hercules standarda) daje spisak desetak ranije otkucanih DOS komandi koje možete ispraviti i ponovo izvršiti gotovo kao da se nalazite u ekranskom editoru. Obzirom da smo navikli na CED, ova koncepcija nam se učinila čudnom ali se brzo pokazalo da je najčešće dosta zgodno da se odjednom vidi više prethodnih naredbi pošto se tako mnogo lakše pronalazi ona koja je trenutno potrebna. Na žalost, PCKKEY ne podržava sinonime koji su jedna od udarnih karakteristika CED-a pa smo njegov editor DOS komandi morali da isključimo (srećom, postoji opcija /M za to, kao uostalom i za sve drugo što možete poželeti da konfigurišete) – tako uzima manje memorije i dalje ubrzavajući kurzore.



Pomenimo i PCKSCRN koji ubrzava ispis na ekran – na našem Herculesu bez PCKSCRN-a video komponenta RI testa daje 28.95 sekundi a sa PCKSCRN-om 23.62, što je dobitak od dvadesetak odsto. Posebno je zgodno što PCKSCRN „pamti" nekoliko prethodnih ekrana teksta koji je „odskrolovao" – ovaj tekst možete da pregledate po pritisku na Scroll Lock. Uzevši u obzir sva ubrzanja koja paket nudi, pregled teksta koji se prebrzo pojavio na ekranu itekako dobija na značaju.

### ŠKOLSKI PRIMER

I na kraju – *printer spooler*. Odmah da kažem da nisam ljubitelj ovakvih programa. Pre svega, nisam baš toliki *workaholic* da ne mogu da podnesem što dok štampač nešto ispisuje ne mogu dalje da radim. Drugo, svako ispisivanje zahteva kontrolu i pažnju zbog mogućih zaplitanja papira a osim toga (čak i ne laseru) pravi buku koja smanjuje koncentraciju. Treće i najvažnije, ako se u bafer natrpa mnogo podataka pa onda dođe do zaplitanja papira, nastupa katastrofa pošto su mnogi slogovi u bazi podataka koja se (recimo) obrađuje označeni kao odštampani a u stvari su se „izgubili" negde u memoriji. Zato najviše volim štampače sa baferom od 80 znakova. Pa ipak, ako bih jednom koristio *printer spooler*, to bi svakako bio PCKSPL.COM iz paketa *PC Kwik Power Pak* – program ima toliko opcija za rezidentno i neposredno startovanje i kontrolu štampača da mu je teško odoleti. Uostalom, najveći prostor u uputstvu je posvećan baš ovoj komponenti paketa.

Kada pominjemo uputstvo, dodajmo da se radi o školskom primeru kako treba pisati dokumentaciju jednog uslužnog paketa. Na početku je svaki program opisan ukratko uz nabrajanje opcija koje su zaista potrebne, a na kraju su detaljno opisane sve opcije za slučaj da vam neka od njih u nekoj specifičnoj prilici zatreba. U okviru uputstva je, osim toga, popularno objašnjeno ne samo kako se pojedine komponente paketa koriste nego i šta koja omogućava, na kojim se principima njeno delovanje zasniva, kada treba koristiti RAM disk a kada keš i tome slično. Sve u svemu, knjižica možda i nije toliko atraktivno štampana, ali predstavlja veoma dobro i korisno štivo za čitanje.

A što se tiče zamerki? Gotovo ništa – uz pomenuti nedostatak sinonima u PCKKEY-u, pomenimo veličinu programa SUPERPCK – preko 50K. Po instalaciji u memoriji ostaje svega 17 kilobajta, ali početna veličina programa sprečava njegovo „podizanje" u *high* memoriju primenom QEMM-a. Tako SUPERPCK „troši" 17K osnovne memorije dok PC-CACHE na 386 mašini neće „potrošiti" ni bajt. Ali, niko nije savršen.

Ostaje mi samo da kažem ono što je svakako očigledno iz čitavog ovog prikaza – još od dana kada je pristigao *Opt Tech Sort* nisam ovako uživao u testiranju nekog programskog paketa i mogu ga sa zadovoljstvom preporučiti svim čitaocima – zvaničnih $130 (često i dvostruko manje kod dilera) je zaista mala suma za konačno rešenje problema keširanja vašeg diska!

### LIČNA KARTA

**Program** PC Kwik Power Pak
**Verzija** 1.57, 7. jun 1990
**Namena** Uslužni programi za ubrzanje računara
**Sadržaj paketa** 360K disketa i uputstvo
**Hardversko-softversko okruženje** IBM PC/XT/AT ili kompatibilan, EMS RAM i hard disk poželjni
**Zauzet prostor na disku** najviše 500 K
**Ulaz** tastatura
**Izlaz – ekran** EGA, VGA, Hercules
**Dokumentacije** 270 A5 strana
**Proizvođač** Multisoft Corporation
15100 SW Koll Parkway
Beaverton, OR 97006
USA
Fax (503)646 8267
**Cena** $130

# STROGO PO PROTOKOLU

**U ovom broju bavimo se tehnikama i metodama koje se koriste za što pravedniju i efikasniju podjelu propusnog opsega kanala lokalne mreže među stanicama.**

Dragutin Vuković

Evo, ovo je Zakon Džungle – star i istinit kao svijet. Vuk koji ga poštuje živi. Vuk koji ga krši će mrijet.

Zakon prepliće Džunglu kao puzavice struk. Jer snaga Vuka je Čopor, a snaga Čopora je Vuk.

Kad Čopor sretne Čopor u Džungli, i jedan drugome put prijeći, lezi dok vođe govore, možda prevladaju razumne riječi.

Lovi za sebe, ženku i mladunčad koja hranu čeka. Ali nikad ne ubij iz zabave, i sedam puta nikad – ne ubij Čovjeka.

(„Druga knjiga o džungli", R. Kipling)

Ovo je samo dio pravila koja čine Zakon Džungle. U njemu ima i mnogo drugih zanimljivih pravila za koja bismo poželjeli da ih se pridržavaju i stanovnici Civilizacije. No, ovdje Kiplingovi stihovi služe samo kao ilustracija pojma kojim ćemo se baviti u ovom nastavku. U džungli lokalnih (i ne samo lokalnih) mreža, isprepletenoj puzavicama kablova, naseljenoj raznim zvjerima (računalima, programima, no da i korisnicima), kukcima i bubama (programskim), potrebno je itekako imati neke zakone i pravila ponašanja, inače bi bilo teško preživjeti. Takva pravila koja određuju način na koji se u nekoj mreži mogu upotrebljavati njene funkcije nazivamo protokolima.

Pojam protokol najranije se odnosio na skup pravila ponašanja u određenim situacijama na carskim i kraljevskim dvorovima, da bi kasnije postao pojam općenito pravila ponašanja. Za razliku od zakona, nepoštivanje protokola neće direktno uzrokovati kažnjavanje, već samo nemogućnost ostvarivanja željene funkcije. Tako, na primjer, bez smokinga ne možete doći na dodjelu Nobelove nagrade, bez obzira što ste ju zaslužili rintajući tridesetak godine u izlizanim trapericama. Isto tako, ako u paketnu mrežu pošaljete paket koji nema propisano zaglavlje, on neće stići na odredište, bez obzira što uredno plaćate pretplatu i modem vam je posve ispravan. No, za utjehu, ne morate se bojati da će vas pozvati na sud i poslati u zatvor zbog upotrebe nepropisnog zaglavlja paketa u mreži toj-i-toj. (Ne znam što bi na to rekao Josef K.?)

## GDJE, TKO I KOLIKO?

Kao što je već rečeno, u lokalnim mrežama se koristi tehnika kanala s višestrukim pristupom. U ovom nastavku bit će govora o protokolima koji služe za upravljanje kanalom, dodjeljivanje propusnog opsega kanala mrežnim stanicama i određivanja prava pristupa stanice u kanal.

Pitanje upravljanja kanalom svodi se na odluku da li će funkcije protokola za pristup i dodjeljivanje propusnog opsega kanala biti centralizirane u jednom čvoru mreže ili distribuirane u svim čvorovima. U suvremenim mrežama izbor pada na distribuirane protokole, ponajviše zbog toga da rad čitave mreže ne bi ovisio o radu tog centralnog čvora, a i zbog raznih drugih razloga koji su već spominjani.

Problem dodjeljivanja propusnog opsega kanala sastoji se u tome koliki dio propusnog opsega kanala će pripasti pojedinoj stanici, da bi se podjela obavila što efikasnije i ravnopravnije.

Protokoli za pristup kanalu određuju koja od mrežnih stanica će u određenom trenutku imati pravo emitiranja poruke u kanal.

Dakle, odgovori na ova tri pitanja određuju GDJE je locirano upravljanje kanalom, TKO ima pravo pristupa kanalu i KOLIKO od kapaciteta kanala će iskoristiti.

O smještaju funkcija upravljanja pristupom u mrežu ne treba puno raspravljati. Razlozi za izbor distribuiranog upravljanja kanalom lokalne mreže tako su očigledni da se danas više niti ne razmišlja o centraliziranom upravljanju (vidi prethodni broj „Računara").

## PODIJELI PA VLADAJ

Problem podjele propusnog opsega kanala nešto je složeniji. Pošto se mnogo stanica natječe za upotrebu kanala, kapacitet ili propusni opseg kanala mora se razdijeliti među njima tako da se osigura najefikasnije iskorištenje raspoloživog kapaciteta. Ovom podjelom se u stvari unutar jednog kanala velikog kapaciteta ostvaruje više kanala manjeg kapaciteta za pojedinačne stanice. Tehnika podjele kanala u potkanale naziva se multipleksiranje kanala (engl. channel multiplexing). Tehnike multipleksiranja kanala razvijene su još za potrebe telefonskih komunikacija, a računarska tehnika ih je preuzela i usavršila za potrebe povećanih zahtjeva računarskih komunikacija. Postoje tri osnovne tehnike multi-pleksiranja zasnovane na dijeljenju prostora, frekvencije i vremena.

Multipleksiranje dijeljenjem prostora je najstarija tehnika multipleksiranja. Više fizički zasebnih potkanala (npr. parica) poveže se u zajednički kanal (kabel) i svakoj stanici se dodijeli po jedan od potkanala. Na ovaj način se povezuju telefonski aparati na telefonsku centralu. Svaki telefon ima svoju paricu u kabelu, svoj fizički prostor. Prednost prostornog multipleksiranja je da imamo potkanal uvijek na raspolaganju za upotrebu, a glavni nedostatak je malo iskorištenje propusnog opsega kanala. Ako telefonirate ukupno 3 sata na dan, što je vrlo mnogo, linija vašeg telefona je iskorištena 12.5%, što je vrlo malo. No troškovi linije su isti, bila ona iskorištena 100% ili ništa.

Multipleksiranje dijeljenjem frekvencije omogućuje da se unutar jednog fizičkog kanala frekventni opseg podijeli po frekvencijama na potkanale. Svaki od ovih kanala se može zatim koristiti kao nezavisni kanal za komuniciranje. Ovaj način multipleksiranja koristi se u telefonskoj mreži za vezu između centrala. U radio i TV sistemima ovo je uobičajen način korištenja prijenosnog medija. Podešavanjem prijemnika na određenu stanicu koju želimo slušati u stvari biramo jedan od frekvencijski multipleksiranih kanala. Koncepcijski, frekvencijsko multipleksiranje slično je prostornom, jer se svi potkanali mogu koristiti istovremeno. Nedostatak ovog načina je da se digitalni signali ne mogu prenositi takvi kakvi jesu, već se moraju pretvoriti u analogne i transponirati po frekvenciji u odgovarajući potkanal. Sklopovi za obavljanje tih funkcija su složeni i skupi, pa nisu primjereni upotrebi u lokalnim mrežama.

Multipleksiranje dijeljenjem vremena je najnovija tehnika u računarskim komunikacijama, iako sama ideja nije naročito nova i na razne načine se upotrebljava odavno. Jedan od primjera vremenskog multipleksa je kad sa kolegom na poslu dijelite zajednički telefon. Dok razgovarate, čitav propusni opseg telefonske linije vam je na raspolaganju ali, naravno, vaš kolega ne može telefonirati. Kad završite razgovor, kolega može uzeti telefon i nazivati po želji. Kako će se ravnopravno razdijeliti propusni opseg vašeg telefona ovisi o dogovoru (protokolu) između vas i kolege s posla. U razvijenijem obliku vremenskog multipleksa poruke iz više stanica isjeckaju se na manje dijelove (pakete), pa se naizmjence šalju kroz kanal paketi iz raznih stanica. Pri tome svaki paket upotrebljava puni propusni opseg komunikacijskog kanala, ali u kratkom vremenu. Na prijemnim stanicama se paketi ponovo sastavljaju u čitave poruke. Ovaj način podjele vremenskog opsega najpogodniji je za primjenu u lokalnim mrežama iz više razloga: digitalni signali se prenose u osnovnom obliku, iskorištenje kanala je veliko, lako se prilagođuje promjenjivim zahtjevima opterećenja itd.
Međutim, kod vremenskog multipleksa javljaju se i dodatni problemi. Najznačajniji od njih je problem određivanja kad stanica ima pravo da pošalje svoj paket u kanal i koliko dugo smije kanal koristiti, tj. koliko dugačak smije biti paket. Ovaj problem doveo je do razvijanja čitave nove kategorije protokola nazvanih „protokolima za pristup kanalu".

Treba napomenuti da se često vremensko multipleksiranje koristi u kombinaciji sa prostornim ili vremenskim multipleksom. Telefon-dvojnik primjer je kombinacije prostornog i vremenskog multipleksa, a teletekst je primjer kombinacije frekvencijskog i vremenskog multipleksa. No, takve kombinacije se uglavnom ne koriste u lokalnim mrežama.

## SAMO PO PROTOKOLU

Pošto gotovo sve lokalne mreže koriste ideju komunikacijskog kanala s višestrukim pristupom, paketni prijenos podataka i vremensko multipleksiranje, nije čudo da se uz njih najčešće vezuje i pojam protokola za pristup kanalu. Protokoli iz ove klase imaju ulogu da osiguraju pravedno i efikasno vremensko dijeljenje propusnog opsega kanala među stanicama koje sudjeluju u mreži. Prvi takav protokol ČISTA ALOHA opisan je u prvom dijelu ove serije. Od pojave ALOHA-e do danas nastale su na stotine protokola za pristup kanalu sa razli-




čitim karakteristikama, što nije čudno jer je to ključan element za performanse mreže kao što su efektivna brzina prijenosa i iskorištenje kanala. Opisivanje makar samo značajnijih protokola iz ovog razvojnog niza oduzelo bi previše prostora. Za razumijevanje materije o mrežama ipak je važno znati nešto o tehnikama koje se tu koriste. Stoga ćemo načiniti jedan taksonomski pregled (pregled po svojstvima) klasa protokola za pristup kanalu.

Kod svih komunikacijskih sistema koje svrstavamo u grupu lokalnih mreža javlja se isti problem: veći broj prostorno raspršenih stanica želi komunicirati odašiljajući poruke jednim jedinim komunikacijskim kanalom s višestrukim pristupom. Taj kanal je JEDINO SREDSTVO KOMUNICIRANJA među stanicama i ima svojstvo da može prenositi samo JEDNU PORUKU u određenom trenutku (vremenski multipleks). Ako se dvije ili više poruka odašilja istovremeno u kanal, te poruke će međusobnmo utjecati tj. uzajamno će se oštetiti i niti jedna neće biti ispravno primljena na odredištu. Prisjetimo se: komunikacijski kanal na kojem se razvija ovakva situacija naziva se kanal s višestrukim pristupom (engl. Multiple Access Channel), a okoliš tog kanala je okoliš s višestrukim pristupom (engl. Multiple Access Environment). Sve stanice mogu istovremeno primati poruku koju odašilja jedna od stanica. Poruka se širi difuzno, slično kao kod, na primjer, radio difuzije gdje jedan odašiljač odašilja radio-valove, a svi (uključeni) prijemnici ih istovremeno primaju. Stoga se često u literaturi susreće i izraz difuzna okolina. Ima mnogo primjera okoliša s višestrukim pristupom u svakodnevnici.

Najjednostavniji je primjer grupe ljudi koji razgovaraju. Ako više ljudi govori istovremeno, rezultat je da niko nikoga ne razumije. Protokol koji se u ovom slučaju zove bonton ili lijepo ponašanje određuje da samo jedan čovjek govori a ostali slušaju i ne prekidaju ga dok on ne završi. Tek kad on zašuti, počeli će govoriti netko drugi. Započne li govoriti dvoje istovremeno, jedno od njih će ušutjeti i dozvoliti onom drugom da govori. Koji od njih će dobiti pravo govora određuje posebni skup pravila: muškarac će dati prednost ženi, mlada osoba će ušutjeti pred starijom i slično. Dakle, u određenom vremenskom intervalu samo jedna od stanica može uspješno odašiljati poruku u kanal pa stoga mora postojati način koordinacije pristupa kanalu. Distribuirani algoritam koji stanice izvode da bi odredile pravo pristupa komunikacijskom kanalu naziva se protokol pristupa kanalu (engl. Channel Access Protocol) ili protokol za kontrolu pristupa prijenosnom sredstvu (engl. Media Access Control Protocol).

Algoritam dijeljenja kanala među stanicama kompliciraju dvije notorne činjenice:

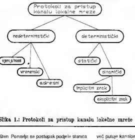
- **REKURZIVNOST**. Stanice su prostorno raspršene i moraju na neki način razmjenjivati informacije potrebne za koordinaciju diobe kanala. Jedino sredstvo komunikacije među stanicama je taj isti kanal. Dakle, da bi koordinirale upotrebu kanala, stanice moraju upotrebljavati taj kanal. Problem je očito rekurzivan.
- **KAŠNJENJE**. Također, zbog raspršenosti stanice ne mogu poznavati točno stanje ostalih stanica u mreži, već samo stanje od prije nekog vremena. Informacija koju stanica A ima o stanju stanice B stara je najmanje toliko koliko je potrebno da poruka stigne od stanice B do stanice A.

Prema načinu na koji se rješavaju ova dva problema možemo međusobno razlikovati protokole višestrukog pristupa. Na taj način dobije se taksonomsko stablo prikazano slikom 1. Protokole najprije možemo podijeliti na dvije opće klase: determinističke i nedeterminističke.

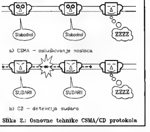
## BACANJE NOVČIĆA

Nedeterministički protokoli (ponekad se nazivaju i stohastički) prepuštaju dosta toga slučaju. Oni rade tako da istovremeno ponude pravo na pristup kanalu grupi stanica, nadajući se pritom da će samo jedna od njih biti pripravna tj. imati spreman paket za slanje u kanal. Pretpostavlja se da stanica koje nema pripremljen paket za slanje neće niti pokušati poslati nešto po kanalu. Protokol tako podijeli skup stanica u dva podskupa: podskup omogućenih stanica (onih kojima je ponuđeno pravo pristupa) i podskup onemogućenih stanica (onih koje to pravo nemaju). Za obavljanje podjele, odnosno određivanje koja stanica pripada kojem podskupu upotrebljavaju se kriterija vjerojatnosti odašiljanja, vrijeme generiranja paketa i adresa stanice. Prema tim kriterijima se dalje dijele nedeterministički protokoli na vjerojatnosne, vremenske i adresne. Nakon obavljene podjele stanica na dva podskupa mogu nastati tri situacije:

a) U podskupu omogućenih stanica nema niti jedne pripravne. Niko ne odašilja poruku u kanal i on u tom intervalu ostaje neisko-




**Slika 1.: Protokoli za pristup kanalu lokalne mreže**




**Slika 2.: Osnovne tehnike CSMA/CD protokola**

šten. Ponavlja se postupak podjele stanica.

b) U podskupu omogućenih stanica pripravna je točno jedna stanica. Ona počinje slanje paketa, i pošto joj niko ne smeta, obavlja prenos do kraja. Nakon toga se ponavlja postupak.

c) U podskupu omogućenih stanica pripravno je više od jedne stanice. One počinju slati svoje pakete i njihovi signali u kanalu se sudaraju. Ako stanice imaju mogućnost detekcije tog sudara, prekida se odašiljanje odmah, i podskup omogućenih stanica se dalje dijeli na dva dijela u nastojanju da se izolira samo jedna pripravna stanica. Ova dioba se rekurzivno nastavlja sve dok se uspješno ne prenese paket iz samo jedne stanice. Ako nema mogućnosti detekcije sudara (engl. Collision Detection), sudareni paketi se emitiraju do kraja i ponavlja se podjela osnovnog skupa stanica. Sudareni paketi ne mogu biti ispravno primljeni na odredištu, pa će se morati ponovo odašiljati. Stoga se upotrebom detektora sudara smanjuju periodi jalovog korištenja kanala i povećava njegovo iskorištenje, a time i efektivna brzina prijenosa.

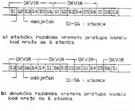
Protokoli s vjerojatnosnim mehanizmom diobe stanica najstarija su i najbrojnija klasa nedeterminističkih protokola. Prvi i najjednostavnijih od njih je već spomenuti protokol ČISTA ALOHA. Kod ALOHA protokola stanica šalje paket u kanal i zatim čeka da vidi da li je stigao na odredište ili nije. Ako nakon nekog vremena ne dobije potvrdu da je paket stigao, zaključuje da nije stigao, pa ga šalje ponovo. Trenutak u kojem će ponoviti slanje paketa bira se slučajno, da se izbjegne mogućnost ponovnog sudara sa istom stanicom. To međutim, ne isključuje mogućnost sudara sa nekom drugom stanicom, koja je u međuvremenu postala pripravna za odašiljanje. Kasnije je nastala varijanta ovog protokola nazvana ALOHA sa vremenskim odsječcima (engl. Slotted ALOHA). Vrijeme je razdijeljeno u intervale ili odsječke jednake trajanju jednog paketa. Stanica smije početi odašiljati samo na početku vremenskog odsječka. Time je dvostruko smanjen period ranjivosti paketa i povećana vjerojatnost uspješnog odašiljanja. ALOHA ne ispituje situaciju na kanalu prije nego počne odašiljati te tako ne zna da li neki paket već putuje kanalom. Netko se dosjetio da prije početka slanja paketa „oslušne" da li na kanalu postoji osnovni signal „nosilac" znak da je kanal u upotrebi (Slika 2 a).

Tako je nastala klasa protokola prozvanih protokolima višestrukog pristupa s osjećanjem nosioca (CSMA, od engl. Carrier Sense Multiple Access). Dva ranija protokola ove vrste su neustrajni CSMA (engl. nonpersistent CSMA) i p-ustrajni CSMA protokol (engl. p-persistent CSMA). Kod neustrajnog CSMA protokola stanica, nakon što postane pripravna, ispituje kanal. Ako ne detektira nosilac na kanalu (što znači da niko ne koristi kanal) počinje sa slanjem paketa. Ako je kanal zauzet (postoji signal nosilac), tada se slanje paketa odgađa za srednja vremena prema nekom lokalnom algoritmu za određivanje vremena odgađanja. Kad istekne vrijeme odgađanja, stanica ponovo ispituje kanal kao da je u tom trenutku postala pripravna. Kod p-ustrajnog protokola stanica, nakon što postane pripravna, ispituje kanal. Ako je kanal zauzet, stanica i dalje osluškuje kanal čekajući da se završi odašiljanje paketa koje je u toku. U trenutku kada kanal postane slobodan stanica počinje odašiljati sa vjerojatnošću p, a sa 1-p odgađa slanje paketa za neki vremenski interval. Ako je slanje odgođeno, po isteku vremena odgađanja ponavlja se postupak ispitivanja kanala.

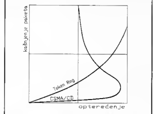
Protokol za pristup kanalu kod mreže Ethernet najpoznatiji je primjer 1-ustrajnog CSMA (p-ustrajni CSMA sa p=1). Postoji više različitih metoda za određivanje vremena odgađanja. Najpoznatija je metoda binarnog odstupanja (engl. Binary Backoff) primijenjena također kod mreže Ethernet. Vrijeme odgađanja je slučajna varijabla s jednolikom distribucijom na fiksnom vremenskom intervalu, tj. ista je vjerojatnost da se izabere bilo koje vrijeme iz tog intervala. Nakon svakog ponovljenog sudara interval se udvostručuje. Komercijalna varijanta Etherneta koristi modificiranu metodu pod nazivom ograničeni binarno eksponencijalni algoritam odstupanja (engl. Bounded Binary Exponential Backoff Algorithm). Vrijeme odgađanja u n-tom pokušaju je jednoliko distribuirana slučajna varijabla u intervalu od 0 do $2^{n-1}$ (n-1) jedinica vremena, za pokušaje od 0 do 10, za pokušaje od 11 do 15 (dozvoljeno je najviše 16 pokušaja), interval je ograničen na 0 do 1023 vremenske jedinice. Jedinica vremena je ovdje predstavljena trajanjem 512 bita na kanalu, što uz frekvenciju odašiljanja od 10MHz iznosi 51.2 mikrosekunde.

Protokoli iz ove klase često radi povećavanja efikasnosti koriste razne metode detekcije sudara na kanalu, pa tako nastaju protokoli sa osjećanjem nosioca i detekcijom sudara (CSMA/CD, Carrier Sense Multiple Access with Collision Detection). Pored nekih složenijih i manje pouzdanih metoda (npr. mjerenje snage signala, mjerenje srednje razine struje i sl.) najpoznatija je metoda „slušaj-dok govoriš" (engl. listen while talk). Ideja je preuzeta od analogije razgovora grupe ljudi: svaki čovjek dok govori istovremeno i sluša. Ako ono što čuje nije identično onome što govori, zaključiti će da se nečiji govor miješa s njegovim, tj. da je na kanalu nastao sudar. Ista metoda se primjenjuje kod lokalne mreže Ethernet: stanica istovremeno odašilje i prima signal, primljeni signal se uspoređuje s odaslanim — ako postoji razlika, nastao je sudar (Slika 2 b).

Nedeterministički protokoli sa vremenskim mehanizmom određuju skup omogućenih stanica na osnovi vremena u kojem su pojedini paketi generirani, tj. u kojem su stanice postale pripravne. Protokol pamti trenutak t0 odabran tako da su svi paketi koji su generirani prije t0 već poslani, a niti jedan paket generiran nakon t0 još nije odaslan. U najjednostavnijoj izvedbi ovog protokola vrijeme je podijeljeno na odsječke duljine trajanja jednog paketa. Nakon što detektiraju slobodan kanal, sve stanice odabiraju početni vremenski prozor veličine W i odašiljaju ako imaju paket u generiran u intervalu (t0, t0+W). Ako niti jedna stanica nema paket generiran u tom intervalu, kanal ostaje neiskorišten u tom intervalu. Vrijednost t0 se mijenja u t0+W i postupak se ponavlja. Ako točno jedna stanica ima paket generiran u tom intervalu, ona ga šalje bez sudara. Po završetku odašiljanja t0 se mijenja u t0+W i postupak se ponavlja. Ako više od jedne stanice ima generirani paket u tom intervalu, nastaje sudar, tada se prozor smanjuje na polovinu W/2 i sa postupak se ponavlja sve dok u prozoru ne ostane samo jedna stanica. Jedan od poznatijih protokola iz ove klase je CSMA s prividnim vremenom (engl. Virtual Time CSMA). Nedostatak ovih protokola je da je potrebno na neki način globalno sinhronizirati satove u stanicama da bi vrijeme t0 uvijek bilo jednoznačno




**Slika 3.: Dijeljenje vremena pristupa kanalu**




**Slika 4. Kašnjenje paketa kod CSMA/CD i Token Ringa**

određeno

Nedeterministički protokoli s adresnim mehanizmom upotrebljavaju adrese stanica pri određivanju podskupa omogućenih stanica. Skup adresa svih stanica uređen je tako da čini uravnoteženo binarno stablo. U početku podskup omogućenih stanica uključuje čitavo stablo, tj. sve stanice imaju pravo pristupa prijenosnom mediju. Nakon pojave prvog sudara na kanalu podskup omogućenih stanica se reducira tako da se iz njega isključi jedno od dva podstabla. Postupak reduciranja se nastavlja nakon svakog ponovljenog sudara, sve dok u podskupu omogućenih stanica ne ostane samo jedna pripravna stanica. Koje od podstabala će se isključiti određuje se slučajnim izborom kako bi se obezbijedilo što ravnopravnije dodjeljivanje prava pristupa. Postoje razne varijante ovog osnovnog protokola, ali svi sadrže isti nedostatak: svaka stanica mora raspolagati tablicom adresa svih ostalih stanica aktivnih u mreži. To prilično komplicira postupak kad se nova stanica uključi u mrežu (ne misli se na fizičko priključivanje, već na jednostavno uključivanje stanice kad ujutro dođete na posao).

## NE HAZARDIRAJMO!

Osnovni cilj izumitelja determinističkih protokola bio je da izbjegnu sudare na kanalu, tj. da ništa ne prepuste slučaju. Jedini način da se to ostvari bio je da se unaprijed, bar za neki kraći interval vremena, odredi (determinira) redoslijed odašiljanja paketa iz pripravnih stanica. Ovaj redoslijed se može odrediti statički, jednom za svagda, ili dinamički, za svaki naredni period iznova. Prema tome, ova se klasa i dijeli na protokole sa statičkom razdiobom i protokole s dinamičkom razdiobom.

Deterministički protokoli sa statičkom razdiobom najčešće dijele vrijeme pristupa kanalu ciklički među stanicama. Vrijeme je podijeljeno na periode nazvane okvirima. Svaki okvir je dalje podijeljen na odsječke. Odsječaka u okviru ima toliko koliko je stanica u mreži. Stanica ima pravo odašiljanja paketa u svom odsječku u svakom okviru (Slika 3.). Ako stanica nema paket za odašiljanje, njen odsječak vremena prolazi neiskorišten. Zamislimo li, na primjer, mrežu sa 100 stanica u kojoj samo dvije stanice razmjenjuju podatke, ukupno iskorištenje



kanala će biti svega 2%. Iako ostale stanice nisu čak ni uključene, dvije aktivne stanice ne mogu iskoristiti dio kapaciteta kanala koji pripada neaktivnim stanicama. Stoga su ovi protokoli izrazito neefikasni i danas ih nitko i ne upotrebljava.

Deterministički protokoli sa dinamičkom razdiobom nastali su kao rezultat napora da se poveća efikasnost protokola sa statičkom razdiobom. Osnovna ideja je pri tome bila da se pravo pristupa kanalu ne dodjeljuje neaktivnim stanicama. Vrijeme je opet podijeljeno na okvire (koji više nisu fiksne duljine), ali se u svakom okviru odsječak dodjeljuje samo stanicama koje imaju pripravan paket za odašiljanje. Tako su izbjegnuti neupotrebljeni vremenski odsječci i efikasnost protokola je veoma povećana. Ipak se nešto vremena troši na određivanje redoslijeda slanja paketa i prenošenje prava pristupa kanalu. Pravo pristupa kanalu dodjeljuje se stanici tako da joj se pošalje neki „znak" (engl. token). Ovaj znak može biti izričit (eksplicitan), tako da se stanici pošalje paket sa porukom „imaš pravo pristupa kanalu u sljedećem odsječku", ili prikriven tako da stanica iz određenih događaja na kanalu saznaje kad nastupa njeno pravo slanja paketa.

Tako se i ovi protokoli dalje dijele na protokole s eksplicitnim znakom i protokole s implicitnim znakom. Protokoli s eksplicitnim znakom su jednostavniji i danas se mnogo koriste. Dva od ovih protokola postali su i standardi protokola za pristup kanalu kod lokalnih mreža: protokol s eksplicitnim znakom na sabirničkoj topologiji (standard IEEE 802.4 ili Token Bus) i protokol s eksplicitnim znakom na prstenastoj topologiji (standard IEEE 802.5 ili Token Ring). Protokoli s implicitnim znakom su složeniji, često zahtijevaju dodatne sklopove i uređaje. To su relativno mladi protokoli i nastali su u nastojanju da se pronađe protokol koji će omogućiti rad lokalne mreže u realnom vremenu. Osnovni zahtjev koji treba zadovoljiti da bi mreže mogla raditi u realnom vremenu je da vrijeme koje protekne od trenutka generiranja paketa do trenutka kad paket stigne na odredište (tzv. kašnjenje paketa, engl. packet delay) bude ograničeno. Što znači da, ako za neku mrežu kažemo da ima kašnjenje paketa T, onda to vrijeme nikad ne smije biti premašeno.

Pitanje kašnjenja paketa upravo je ključno pitanje za odlučivanje da li odabrati deterministički ili nedeterministički protokol. Na slici 4 može se vidjeti ovisnost kašnjenja paketa o opterećenju mreže (broju paketa u jedinici vremena, normirano na maksimalni kapacitet kanala) za dva karakteristična protokola iz grupa determinističkih (Token Ring) i nedeterminističkih (CSMA/CD) protokola. Može se vidjeti da CSMA/CD ne zadovoljava kriterij rada u realnom vremenu, jer kašnjenje paketa nije ograničeno, tj. moguće je čak beskonačno kašnjenje (paket nikada ne stigne) pri jako velikom opterećenju. Token Ring protokol, naprotiv, nikad neće imati kašnjenje paketa veće od Tm. Rad mreže u realnom vremenu je uvjet za neke primjene mreže, npr. u prilagodljivim proizvodnim sistemima (engl. Flexible Manufacturing Systems, FMS) ili prijenosu govora. S druge strane, međutim, protokol CSMA/CD pri manjim opterećenjima pokazuje puno manje kašnjenje paketa od Token Ring protokola, pa time ima i veću efektivnu brzinu prijenosa. Možemo samo zaključiti da će općenito deterministički protokoli biti pogodniji za mreže sa stabilno velikim opterećenjem i uvjetom ograničenja kašnjenja paketa, dok će nedeterministički protokoli biti efikasniji kod mreža sa vrlo promjenjivim opterećenjem.

**U sljedećem broju**
**ELEMENTI DIZAJNA LOKALNE MREŽE**
**3. dio**
**Topologija i protokoli**
**za pristup kanalu komercijalnih lokalnih mreža**
**Ethernet, Token Ring, Token Bus.**

# NUMERIČKI MITOVI I LEGENDE

**Nema te matične ploče na koju nije ugrađeno i podnožje za tzv. matematički koprocesor, ali je to u personalnim računarima još uvek izuzetno redak deo hardvera. Sve je, međutim, više programa koji se u matematičkim operacijama oslanjaju na usluge koprocesora. Neki medu njima se bez koprocesora ne mogu čak ni pokrenuti, a drugi naprosto rade 3-10 puta brže. Mi smo na uporednom testu suočili dva popularna i relativno pristupačna modela.**

Dejan V. Veselinović i Bojan Tepavčević

Do sada se radilo o delu tržišta koji se smatrao marginalnim u računarskoj industriji – isuviše ograničenom po broju da bi bio interesantan proizvođačima hardvera. To je, međutim, potpuno netačno, a zeca treba tražiti u sasvim drugom grmu. Naime, iako je Intel prodavao licence za proizvodnju procesora 8086/8088, 80186/80186 i 80286, nikada nikome nije prodao licencu za proizvodnju odgovarajućih numeričkih koprocesora, a ista filozofija važi u proširenom obliku (odnosi se i na same procesore) za sve procesore i koprocesore 32-bitne generacije. Intel je želeo da ostane jedini mogući izvor (tzv. „sole source") koprocesora, tj. svetski monopolista. Istina, pojavili su se daleko bolji (i skuplji) alternativni koprocesori firme Weitek, ali da bi se njihova brzina iskoristila bilo je potrebno platiti bar tri puta više i rekompajlirati programe. To, međutim, nije ni lako ni jeftino, niti programeri vole razvijaju tri podgrupe istog programa.

## FRIZIRANI KLONOVI

Negde sredinom prošle godine pronele su se glasine da čak nekoliko firmi radi na klonovima numeričkih koprocesora, i to ne samo onih 16-bitnih nego i 32-bitnih. Industrija je zabrujala, svi su uhvatili dah i potrčali da kupe karte za sudsku arenu, jer nije bilo ni najmanje sumnje da će Intel tužiti. I zaista, pred kraj godina dve američke firme, Integrated Information Technology (IIT) i Cyrix, su ne samo najavile, nego i prikazale prototipove. Radi se o numeričkim koprocesorima koji imaju potpuno isti raspored nožica kao i originali, što znači da su neposredno zamenljivi, koštaju u dlaku isto koliko i originali, ali sav posao obavljaju prema rečima proizvođača najmanje 30% a negde i 150% brže. Povrh svega, pored standardnih naredbi imaju ugrađene i neke dopunske naredbe koje istina zahtevaju posebne programske pozive (tj. pozive kojih kod Intelovih koprocesora uopšte nema), ali zato mogu obaviti posao i do 80 puta (!) brže od originala pri istim radnim uslovima.

Poučeni prethodnim iskustvima, ovi proizvođači nisu izveli sada već klasičan trik obrnutog projektovanja („reverse engineering") po kome se kupi original, laserom mu se skine omotač, substrat se snimi i jednostavno duplira: da su tako postupili dobili bi kopije originala, a hteli su nešto bolje. Zato, umesto prostog kloniranja, oba proizvođača su se opredelila za kloniranje funkcija, ali ne i tehničkih rešenja. Drugim rečima, hteli su iste funkcije ali brže, što podrazumeva različita rešenja; kada su se već za to opredelili, nije im bilo teško da naprave i dopunski korak u smislu dodavanja novih funkcija i izmene tehnološke baze proizvodnje.

Naime, od projektovanja originalnih koprocesora je prošlo već nekoliko godina, tehnologija je napravila veoma velike korake unapred, a i stvarnost ih je terala na takve korake. Primera radi, jedan koprocesor 80387 na 20 MHz radi oko 6-8 puta brže od standardnog Intelovog modela 80287 na 10 MHz. Iako je osnovna razlika između ovih koprocesora u činjenici da je prvi 32-bitni a drugi 16-bitni, jednako je očigledna i razlika u radnom taktu, mada u slučaju koprocesora ne važi linearan pristup povećanju radnog takta (odnosno, dva puta veći takt ni slučajno neće dati dva puta brže proračune).
Razlog ovome je činjenica da od modela 80287 nadalje koprocesor sve više i više zavisi od procesora za razne operacije, kao što je recimo pristup memoriji, baratanje podacima i slično.

## KLONOVI NA SUDU

Osnovni jaz koji je nastao se odnosio na razlike između naglo napredujuće generacija 32-bitnih mašina i još uvek veoma aktivne i nimalo mrtve generacije 16-bitnika, koja je čak počela da sakuplja snagu da odumre. Harris i AMD su prikazali verzije procesora 80286 prvo na 16 a zatim na 20 MHz, a sada najavljuju i verziju za 25 MHz; pored toga, umesto stare NMOS tehnologije, koriste savremeniju CMOS tehnologiju koja troši manje struje, što ove procesore čini pogodnim za prenosive računare. Logički, proizvođačima koji koriste ove procesore je zatrebala jednako ekonomičnija verzija koprocesora, po mogućnosti i na većim radnim učestanostima.

Prvi proizvođač koji se odazvao ovim zahtevima je bila američka firma IIT, koja je krajem prošle godine najavila i ubrzo počela da prodaje svoje „frizirane" klonove numeričkih koprocesora. Odmah su počeli sa radnim taktovima od 10 i 12 MHz, a uskoro se pojavila i verzija za 20 MHz. U slučaju klonova 32-bitnih koprocesora, odmah su ponudili celu gamu od 20 do 33 MHz. Prema njihovim podacima, svi modeli su pod istim uslovima 30-300% brži od originala, a imaju funkcije (kao recimo baratanje matricama kao celinama) koje original uopšte nema, a sve to u kombinaciji sa stoprocentnom zamenljivošću jednog drugim (kompatibilnost na nivou nožica).

Intel je naravno odmah podneo tužbu i na veselje cele industrije glatko izgubio spor. Naime, IIT-u nije bilo ni malo teško da pokaže da su tehnička rešenja unutar čipova potpuno drugačija od Intelovih, a sud je odmah naglasio da ne prihvata ideju da iko može da patentira princip koprocesora.

Da ovaj mali pregled istorije događaja završimo kratkim pregledom današnje situacije: Cyrix i IIT slobodno prodaju svoje proizvode, a Intel je brže-bolje izbacio novije verzije starih koprocesora pod oznakom 80287XL, koji navodno rade 50-60% brže pri istim uslovima od uzora, standardnog koprocesora 80287. Dođe nam da se pitamo da li bi do ovoga ikada došlo da se nije pojavila konkurencija; mislimo da ne bi, jer monopolistu ama baš ništa ne tera na dalji razvoj i na konkurisanje samom sebi.

## INTEL PROTIV INTELA

Nas je zainteresovala cela ova afera iz više razloga. Prvo, zanimalo nas je da ispitamo kompatibilnost klona, i drugo, želeli smo da proverimo tvrdnje o većim brzinama rada. Opredelili smo se za verzije namenjene procesoru 80286 iz često praktičnih razloga: kod nas i u svetu ova klasa računara apsolutno i relativno dominira po zastupljenosti i drugo, glavna bitka se i vodi između 16- i 32-bitne generacije. Primera radi, ako za recimo 22.000 dinara možete da kupite AT koji će numeriku raditi približno jednako brzo kao i recimo AT/386 na 20 MHz koji košta recimo 44.000 dinara, imate osnova da se dobro zamislite nad ekonomijom takve situacije.

Kao platformu za merenja upotrebili smo jedan Chicony LT 3600 prenosivi računar. Ova mašina ima procesor Harris 80C286 koji radi na 20 MHz, a kao osnov koristi poznati Chips & Technologies NEAT paket čipova, što znači da je sasvim reprezentativna za današnju klasu jačih AT računara. Koprocesor može raditi (što se određuje jednim džamperom) na 6 ili na 12 MHz, efekat koji smo iskoristili da usput ispitamo uticaj radnog takta procesora na numeričke operacije u prisustvu koprocesora.

Zatim smo se snabdeli sa jednim Intel 80287-10 koprocesorom jer na žalost nismo nikako mogli da uđemo u trag propisnoj verziji za 12 MHz i jednim IIT 802C87-12, dakle propisnom verzijom od konkurencije. Interesantno je primetiti da IIT košta DEM 10 manje od Intela, što je naravno zanemarivo na tim nivoima cena, ali je fascinantno ukoliko ovaj korpocesor zaista radi brže od uzora, jer onda ispada da em radi brže, em je jeftiniji za cenu jedne pice u Minhenu.

Za testiranje smo upotrebili naše standardne verzije „Unitest" programa, eli i test, odnosno demo program koji se deli uz IIT-ov koprocesor. Sve rezultate smo merili u svim mogućim varijantama sa oba koprocesora (286 na 10 MHz, 287 na 6 i 12 MHz, 286 na 20 MHz, 287 na 6 i 12 MHz), a u nekim tabelama smo čak radi upoređivanja rezultate i sa jednim originalnim IBM PC računarom opremljenim odgovarajućim koprocesorom – tek da vidimo dokle je dogurala ekonomske klasa. Inače, napominjemo da program koji mi koristimo za testove CAD-a veoma neefikasno koristi koprocesor daleko iznad proseka, što znači da će marginalne razlike kod nas u praksi biti daleko veće u drugim programima, a posebno AutoCAD-u. Najbolji reper efekata je bez ikakve sumnje test unakrsnih proračuna, koji pokazuje sve očekivane efekte ubrzanja.

Odmah na početku smo doživeli mali maler, naime, Intel 80287-10 je naprosto odbio da radi na 12 MHz. Ovo je prvi put da nam se

| Model<br>Test | Chicony LT 3600 Radni takt procesora = 20 MHz | | | |
|---|---|---|---|---|
| | Bez ko. | I287=6 | IIT = 6 | IIT = 12 |
| **TESTOVI IIT-a:** | | | | |
| FLOP/sekundi | -- | 18.808 | 60.610 | 71.754 |
| TPE/sekundi | -- | 2.642 | 7.799 | 10.238 |
| Rotacija žičanog modela, sek. | -- | 97,71 | 38,22 | 31,69 |
| Okretanje u prostoru, sek | -- | 13,74 | 10,49 | 10,17 |
| Unakrsni proračun, sek. | -- | 22,25 | 13,52 | 12,42 |
| Mandelbrot slika, sek. | -- | 360,80 | 68,68 | 49,67 |
| CAD vreme crtanja, sek. | -- | 2,14 | 1,48 | 1,43 |
| **NAŠI TESTOVI:** | | | | |
| Indeks koprocesora (AT=1,00) | -- | 1,10 | 3,50 | 4,50 |
| Kilo Whetstone-a/sekundi | -- | 204,23 | 417,42 | 494,71 |

| Unitest programski testovi | | | | |
|---|---|---|---|---|
| **CAD (DesignCAD 3-D):** | | | | |
| - Crtanje slike (Generate) | 19,44 | 20,32 | 14,10 | 13,61 |
| - Rotiranje slike (Rotate) | 9,90 | 11,27 | 6,66 | 6,34 |
| - Senčenje slike (Shade) | 107,22 | 116,68 | 85,73 | 84,10 |
| - Skrivanje linija (Hide) | 119,74 | 135,33 | 110,04 | 108,49 |
| I N D E K S (256,30 s = 1) | 1,00 | 0,90 | 1,18 | 1,21 |
| **PRORAČUNI (Borland Quattro):** | | | | |
| - Učitavanje radne tabele | 14,76 | 11,36 | 10,14 | 8,91 |
| - Preračunavanje matrice | 116,35 | 51,01 | 35,12 | 33,51 |
| - Nađi i zameni ("1" sa "2") | 153,55 | 72,55 | 45,82 | 43,38 |
| - Pisanje na disk | 3,42 | 2,83 | 2,91 | 2,82 |
| I N D E K S (288,08 s = 1) | 1,08 | 2,09 | 3,06 | 3,25 |
| KUMULATIVNI INDEKS | 2,00 | 2,99 | 4,24 | 4,46 |
| PROSEČNI INDEKS BRZINE | 1,00 | 1,49 | 2,12 | 2,23 |

dogodilo da bilo koji Intelov procesor ili koprocesor odbije da radi na učestanosti većoj od nominalne, ali sa druge strane, ne možemo da se bunimo jer na čipu lepo piše da je predviđen za 10 a ne 12 MHz, a sem toga, i plaćen je kao jeftinija verzija. To odbijanje da radi se ogledalo u potpunom zaglupljivanju celog računara, mada ne i u blokiranju.

Na sreću, LT 3600 omogućava prebacivanje radnog takta sa 12 MHz na 6 MHz, a kada smo to uradili, sve je savršeno radilo. Za razliku od Intela, IIT čip je i deklarisan za brzinu od 12 MHz i sa njim nismo imali nikakvih problema. Druga veoma velika razlika između ova dva proizvoda je u toploti koju otpuštaju radom: nakon 20 minuta rada, Intel je bio toliko vreo da smo morali da koristimo plastične držače, dok je IIT bi malo topliji od mlakog, tek koliko da osećate neku toplotu. To je u osnovi razlika između NMOS i CMOS tehnologije.

## NA BRZINSKOM TESTU

Priložena tabela prikazuje dobijene rezultate. Kao što se lako može videti, IIT je zaista bio brži od uzora u apsolutno svakom pojedinačnom testu. Mi smo (naravno) bili pomalo skeptični prema testovima koje prilaže IIT, uprkos njihovim tvrdnjama da su koristili isključivo Intelove naredbe, ali su rezultati dobijeni na tim testovima manje-više potkrepljeni razlikama koje smo dobijali i u drugim testovima.

Zaključci koji se iz rezultata mogu izvući su višestruki. Prvo, testovi koje IIT prilaže uz svoje koprocesore su u neku ruku „pakovani" u smislu da do maksimuma potenciraju vrline njihovih koprocesora koristeći neobično „čist" kod. U apsolutnom smislu to nije nepošteno jer je činjenica da će sa izuzetno optimizovanim kodom, koji idealno koristi funkcije koprocesora, IIT stvarno raditi mnogo brže od Intela, ali vas ovo može zavesti, jer u praksi malo ko ako iko koristi tako savršeno optimizovan kod. Ovo je dakle slučaj posmatranja istine sa jednog prilično akademskog stanovišta, pa zato i vi ove testove i rezultate shvatite prilično akademski.

Drugo, od obećavanih ubrzanja u radu od 10-30 puta, kako svi prodavci koprocesora vole da navode u reklamama („... do 30 puta brže!") takođe koristi logiku savršeno optimizovanog koda. Programi za unakrsne proračune i za CAD su obično dobro prilagođeni upotrebi koprocesora, mada program za CAD koji mi koristimo veoma slabo koristi prednosti koprocesora. Quattro, sa druge strane, pokazuje značajne dobiti u brzini, pa iako ne radi „do 30 puta brže", radi 3-4 puta brže, što se i te kako oseti. Na većem modelu (iako ni ovaj naš nije mali) proračuna, dobili bi bile čak i veće, pa možete računati na stvarna ubrzanja 4-6 puta. U programima pisanim na recimo fortranu, ubrzanja bi bila daleko veća, gotovo sigurno veća od 10:1. Time odgovaramo na dilemu kupiti koprocesor ili ne: ako se bavite ovim poslom, obavezno ga kupite. AutoCAD daleko bolje koristi koprocesor od našeg DesignCAD-a, koji je usput i starija verzija, a novije verzije AutoCAD-a ni ne možete pokrenuti bez koprocesora; dakle, opet nema nikakve dileme da li ćete kupiti koprocesor ili ne.

Treće, pošto je IIT na doslovice svakom testu pokazao jasne prednosti nad Intelom, i pošto katkada košta čak i nešto malo manje od Intela, nema dileme koji od njih valja kupiti. Iako su razlike manje od reklamiranih i iznose 30-45%, opet za iste pare dobijate bolje performanse, pa ne bi smelo biti dileme oko toga koji od njih kupiti. Eventualne strahove oko kompatibilnosti možemo raspršiti konstatacijom da nakon 3 meseca rada sa IIT 802C87 nismo našli ni na jedan program koji je odbio da radi. Drugi argument bi bio da ako su razni proizvođači uspeli da kloniraju cele računare i niz drugih čipova, nema razloga zašto ne bi jednako uspešno klonirali i ovaj, za današnje pojmove jednostavan čip.

I na kraju, četvrto, IIT koristi CMOS tehnologiju, koja ima dve bitne prednosti nad klasičnom NMOS tehnologijom, a to su manja potrošnja struje i manja količina proizvedene toplote (prvo povlači drugo). Kao stvar principa, manja toplota znači veću pouzdanost i duži životni vek delova. Bez obzira na sve ostalo, prema tome, nedvosmislene prednosti idu IIT-u. Ako pak nameravate da ugradite ovaj koprocesor u prenosivi računar sa akumulatorima, onda vam manja potrošnja i te kako znači, pa tako opet nemate dileme.

Dakle, u sva četiri slučaja nemate dileme. Dodajmo još i to da je priložena dokumentacija IIT-a jednako impresivna kao i ona od Intela, a ima domaćih ponuđača koji su već gotovo u celosti prešli na ove proizvode (recimo, Gema Electronics, a ima i drugih), čime je uklonjen problem nabavke, jer sada Intel morate da tražite. Šteta je jedino što nismo imali na raspolaganju i novu verziju Intelovog 80287 sa oznakom 80287LX, koji po njima treba da radi 50-60% brže, ali ko zna, jednog dana...

## PROCESOR U KOMI

I tu bi trebao da bude kraj ove male priče. Međutim, sasvim slučajno, tokom testiranja smo pokrenuli program za statičke proračune koji je napisao B.T. U okviru tog programa postoji jedno mesto na kome se od koprocesora očekuje da proračuna ASIN (arkus sinus) od 1. Program je inače napisan pomoću *MS Fortran 5.0*, a služi statičkim proračunima. Sve je bilo u najboljem redu do prve pojave ASIN 1, kada nas je koprocesor obavestio da se traženi rezultat nalazi „Out of domain" (van domena dohvata). Kada je ista petlja isprobana na Intel 80287 i 80387 koprocesorima, kao i na IIT 803C387, sve je lepo radilo, ali na IIT 802C287 nije htelo pa bog.

Prva provera se odnosila isključivo na deo programa. Ponovljen je zahtev za proračunavanjem ASIN 1 koristeći čist fortran, sa istim rezultatom. Sve ostale provere su mimo izbacile željeni rezultat. Druga provera se odnosila na funkciju ASIN, kada je broj 1 izmenjen u 0.9999, rezultat je bio kako treba. I početni program je radio onako kako pre unošenja ove izmene nije hteo. Naizgled sitnica, jer izmena broja 1 u 0.9999 će dati funkcionalne iste rezultate. Međutim, postavlja se pitanje kako koprocesor shvata broj 1, kako u ovom tako i u drugim slučajevima: ako 32-bitna verzija čipa iz iste kuće nema problema sa ovim, zašto onda ova verzija ima? Dalje, na tržištu se trenutno nalaze čak tri verzije ovog koprocesora iz tri izvora: Intel, IIT i Cyrix. Za Intel-ov proizvod se zna da je čist (i to smo proverili), IIT izgleda ima problema, a do Cyrix verzije nismo uspeli da dođemo na vreme do zaključenja testa.

Čim smo naišli na problem, poslali smo telefaks firmi IIT, ali do danas (14.10.1990) nismo dobili odgovor, iako smo detaljno objasnili problem.

Za zainteresovane, evo programskog reda kojim smo uspeli da zbunimo koprocesor (*MS Fortran 5.0*):

```
WRITE(*,*) ASIN(1.)
END
```

Probajte i gledajte kako korpocesor pada u nesvest.

Sve ovo nas je navelo da koprocesore shvatimo daleko ozbiljnije no što smo to do sada činili (kada nije bilo izbora – kupite Intel i to vam je što vam je), posebno zato što sve veći broj korisnika računara i kod nas počinje da se oprema ovim skupim, mada dobrim a neretko i nužnim dodatkom.

Ovaj događaj je nužno izmenio naš stav o ovom proizvodu.

# POČETKOM OKTOBRA U ŠEHER SARAJEVU

**Početkom oktobra održani su drugi „Sarajevski računarski susreti". Prošle godine se pričalo o virusima, a ove o računarskoj obradi teksta (obrada teksta, računari u novinarstvu, stono izdavaštvo). Organizator je i ovoga puta bio izuzetno vitalni „PC Pool", pod vođstvom još vitalnijeg Vladimira Videnovića.**

Računarski susreti su zamišljeni kao nekakav mini sajam ili izložba računarske opreme tokom dana i kao okrugli stolovi predveče. Dom mladih u Sarajevu je za potrebe manifestacije odredio dve sale, jednu za izložbu i amfiteatar za okrugle stolove. Ako je sudili po opštoj podršci sredstava javnog informisanja (TV Sarajevo 3 kanal, Radio Sarajevo i NIP „Oslobođenje") sa zadovoljstvom možemo zaključiti da ove manifestacija ima sve šanse ne samo da opstane i postane stalna, već i da napreduje u svakom pogledu.

## JEDAN ALI VREDAN

Od izlagača opreme prisustvovalo je svega četvoro (trebalo je da ih bude više), ali za utehu, radilo se o probranim izlagačima koji su došli dobro opremljeni. Najinteresantnije je to što je organizatoru pošlo za rukom da svi izlažu nešto drugo, dakle da ne konkurišu jedni drugima, a da posetioci opet imaju veoma dobar pregled ponude na našem tržištu. Tako je beogradski Ibis Sys izlagao pet glavnih modela firme Mannesmann-Tally (od MT81 o kome smo pisali do laserskog štampača MT905), ljubljanski Gambit je prikazao tri računara opremljena za razne namene (sa video podsistemima koji su koristili NEC 3D monitor i posebno ETAP monitor namenjen DTP-u), zagrebačka firma Penhard je prikazala impresivnu gamu proizvoda „pasivne bezbednosti" (filtera za ekrane i raznih drugih ergonomskih dodataka) i najzad, nova firma iz Prozora Prony je prikazala spektar proizvoda disketa svih forma te i kapaciteta, kao i niz traka za štampače.

Sve to se nalazilo u jednoj sali lepo složeno, a odziv posetilaca bi se mogao opisati kao relativno probran, jer je većina pokazala dosta visok stepen poznavanja računarske tehnike a to se najbolje videlo po pitanjima.

Što se izložene opreme tiče, daćemo samo kratak pregled. Ibis Sys iz Beograda je prikazao dva modela na koje skrećemo pažnju, MT 131/9, matrični štampač sa 9 iglica i formatom papira A3 i laserski štampač MT 905, očigledno je da je ovaj drugi bio u duhu teme okruglih stolova, a bogami je dobro i poslužio za razne demonstracije, u nečemu što bi se nazvali INTERYUGOSLAVIA sistemom: štampač od Beograda (Ibis Sys), računar i monitor iz Ljubljane (Gambit), filteri iz Zagreba (Penhard), a diskete iz Prozora (Prony). Gambit je doneo tri računara, dva na bazi procesora i80286 i jedan sa i80386 na 25 MHz sa kešom. Pored standardnog modela monitora, bio je tu i jedan NEC 3D koji stvarno ostavlja bez daha i jedan ETAP monitor koji ima oblik A4 stranice. Sam monitor je zaista dobar, ali je problem u tome što ne prikazuje baš celu A4 stranicu u prirodnoj veličini, za šta mu nedostaje centimetar ili dva – šteta.

Iznenađenja su bila na stolu Penhard-a. Pored čitavog dijapazona filtera za ekran, od kojih su nam se neki čak veoma dopali i za koje Penhard može na licu mesta da dokaže

**Dejan V. Veselinović**



da rade svoj posao, videli smo i par za naše uslove egzotičnih dodataka, kao recimo oslonac za laktove i posebno postolje za noge. U prvi mah, poneseni machismom naših prostora, upitali smo se čemu sve to, ali kada smo malo bolje pogledali demonstraciju i na kraju onako na brzinu probali naslon za lakat a posebno postolje za noge, dosta lako smo se uverili da se uopšte ne radi o glupostima, već o veoma vrednim dodacima računarskoj opremi.

## OKRUGLO...

Okrugli stolovi su se održavali u jednom amfiteatru uz učešće niza gostiju. Prvo su tu bili gosti van Sarajeva, od kojih određen broj na žalost nije mogao da dođe, što je velika šteta, a zatim i niz veoma kompetentnih ljudi iz Sarajeva. Domaćin svim stolovima je bio Vladimir Videnović, rukovodilac „PC Pool"-a, koji je svoju dužnost junački obavio.

Prva tema „Računarska obrada teksta" je okupila niz učesnika, a doajen susreta je svakako bio prof. dr Božidar Krstajić, inače poznat (ako mu niste bili studenti) kao autor verovatno definitivne knjige o poznatom tekst procesoru *ChiWriter*. Ta knjiga je sada u trećem ili četvrtom izdanju, a ako koristite ovaj program obavezno je potražite u knjižarama i bar pogledajte, verovatno ćete je i kupiti. Ing. Anil Agović je govorio o programu *Lotus Manuscript*, a vaš izveštač naravno o *WordPerfect*-u. Jedina šteta je bila u tome što nije bilo Duška Savića kao osvedočenog poznavaoca inače odličnog programa *Microsoft Word*. Verovatno najbolje je prošao upravo *WordPerfect*, ne zato što je vaš izveštač bolje izlagao materiju, nego zato što je bio najbolje naoružan, jer je kao osvedočeni čudak pošao na put sa prenosivim računarom, pa je mogao da demonstrira razne aspekte korišćenja.

Druga tema „Računari u novinarstvu" okupila je isto tako eminentnu ekipu. Gosti su bili Milić Petrović, novinar novosadskog lista „Preduzetnik", Elvedin Kanterdžić, novinar sarajevskog „Oslobođenja" i saradnik „Nedjelje" i vaš izveštač, koji je ko zna kako uspeo da bude kandidovan i za ovu temu. Na žalost, ova tema se odvijala na dan utakmice, pa je bila slabije posećena od prve i treće, što je posebna šteta, jer je bilo veoma interesantno čuti iskustva iz novosadskog „Preduzetnika". Takođe je bilo prijatno čuti dokle se sa računarima stiglo i u „Oslobođenju", gde se kao i praktično svugde drugde stvar svodi na nekoliko entuzijasta, kojih jeste malo, ali su prodorni. Ipak, opšti zaključak koji smo stekli iz diskusije je da su novinari dosta brzi na prihvatanju računara, naravno, kada im se lepo pokaže šta sve mogu da urade sa njima.

Treća tema je bila „Stono izdavaštvo", a učestvovali su Anil Agović, Saša Svitlica, domaćin Vladimir Videnović i naravno vaš izveštač, sada već po logici „ajd kad je već tu, da ne uvredimo čoveka". Ovo je bila najposećenija tema, sa dosta pitanja i odgovora, što se vidi i po trajanju, jer se sve odvuklo do dosta kasno naveče. Ujedno, ova tema je izazvala najviše polemike i najviše razmene iskustava kako između učesnika i publike, tako i među samom publikom.

Tokom svih razgovora u okviru svih tema kolege Viktor Glavina iz Ibis Sys-a, Miran Povše iz Gambita i Danilo Pešić iz Penhard-a su objašnjavali efekte i uticaj njihove opreme na date teme. Naravno, tu su bile i brojne demonstracije uz pomoć epidijaskopa, a u okviru demonstracija smo videli i jedan veseo mali događaj. Pred temu „Računari u novinarstvu", ekipa na čelu sa vašim izveštačem je pokušala da nazove *Sezam*, sve je bilo spremno, računar i monitor i epidijaskop, i čak tri modema. I tako, zovemo mi zovemo, ali se *Sezam* ne odaziva. Tek po povratku u Beograd saznamo da smo svo vreme zvali pogrešan broj, odnosno stari broj *Sezama*, to saznamo naravno sa samog *Sezama*, na opšte oduševljenje svih koji su pročitali informaciju.

## ...PA NA ĆOŠE

Sve u svemu „Dani računarstva u Sarajevu" je jedna veoma uspela akcija koju bi na svaki način trebalo negovati. Evidentno je da za ovakve tribine ima interesovanja, ima uslova, a što je najvažnije, ima i dobrog organizatora. Kada jedna relativno mlada grupa kao što je „PC Pool" može da pozove veći broj gostiju od kojih su neki izlagači, i da se odazovu svi koji ikako mogu da dođu (i da onima kojima to nije pošlo za rukom bude žao što su bili objektivno sprečeni), onda se radi o nečemu što bi trebalo podržavati, jer je velika verovatnoća da sve to može da se uradi još bolje.

A naravno tu je i samo Sarajevo (u klasi Cray-a), „Palata" (Seymore Cray da im kafa bude), ćevapi kod Ferhatovića (mainframe klasa), pa Morića Han (486 na 50 MHz) – zamislite, čak smo uspeli da nekih pola sata ne pričamo o računarima.

# LOTUS MISLI OZBILJNO

**Početkom oktobra ove godine u beogradskom hotelu Hyatt održana je zvanična promocija zastupnika čuvene svetske firme Lotus u Jugoslaviji. Zvanični zastupnik, odnosno „ekskluzivni distributer" za Jugoslaviju je novosadska firma MP YUSACO, a pod-distributeri su TP NOLIVEL iz Beograda i SRC iz Ljubljane.**

Dejan V. Veselinović

Promociji su prisustvovali i gosti iz inostranstva, predvođeni gospodinom John O'Harom rukovodiocem prodaje proizvoda Lotus u Evropi. Centrala ove firme za Evropu se inače nalazi u Velikoj Britaniji. Prezentaciji su prisustvovali brojni gosti, a sigurno najbrojnija delegacija je bila ona iz „Računara" sa čak tri predstavnika.

## IGRA SUMNJIVIH BROJKI

Sama prezentacija je bila u celosti na nivou uobičajenih događaja te vrste. Prema podacima ove firme, u Jugoslaviji postoji oko 150 000 računara, a ravnajući se prema svetskim prosecima Lotus je došao do zaključka da oko 97 000 mašina ima nekakav program za unakrsne proračune. Opet prema svetskim normama, oko 70% računara sa takvim programima koristi *Lotus 1-2-3*, pa proizilazi da u Jugoslaviji radi oko 68 000 kopija ovog poznatog programa. Ako pretpostavimo cenu od samo USD 200 po primerku, to bi bio punomasni kolač od USD 1 360 000, što ni za koga nije zanemarivo, pa naravno ni za Lotus. Realnija računica bi bila za oko 50% veća, odnosno sve sume bi se povećale za polovinu. Što se same računice tiče, mi smatramo da je broj kopija prilično preteran, ali čak i tako, nema spora da se radi o pozamašnim sumama koje ne treba ostaviti sitnim.

Velika novost prezentirana prisutnima, pored naravno činjenice da od sada postoji zastupnik, jeste najava nove verzije najpoznatijeg proizvoda ove kuće, programa Lotus 1-2-3 verzija 3.1. Ova verzija se razlikuje od prethodne po nizu detalja i jednoj izuzetno za nas Jugoslovene važnoj stvari, a to je podrška kodne stranice 852, koja sadrži sve naše znake. Drugim rečima, program kupljen u Jugoslaviji će imati programsku pobudu za naša slova, što je dvostruko interesantan podatak, kako sa aspekta naših korisnika, tako i sa aspekta činjenice da se u praksi pokazuje brzina nekih svetskih firmi u njihovim naporima da osvoje nova tržišta. Dakle, ovde možemo samo čestitati Lotus-u na brzini i prilagodljivosti.

Druga stvar koja nas je impresionirala jeste činjenica da je već na samoj prezentaciji rečeno i navedeno da mreža distributera već postoji i da nije ni malo ograničena. Zvanični distributeri se nalaze u Novom Sadu, Beogradu, Ljubljani, Splitu, Zagrebu, Skoplju i Sarajevu, veoma impresivno za jedan novi poslovni sistem, a naravno, i veoma pogodno za postojeće i potencijalne korisnike. Nude se svi proizvodi firme Lotus (*1-2-3 V 3.1 i V 2.2, Magellan, Symphony, Freelance Plus, Graphwriter II, Agenda i Manuscript*). Svi osnovni programi koštaju 6 900 dinara, sem 1-2-3 V 3.0/3.1 koja košta 7 950 dinara i 1-2-3 verzije za OS/2 koja košta 9 450 dinara.

Ko o čemu, mi o cenama. Prvo naše pitanje se odnosilo na cene ponuđenih programa, jer su one kao i obično oko 2-4 puta veće nego u SAD, gde su cene uvek najniže. Dobili smo odgovor da su cene svih programa u SAD uvek najniže u svetu jer se tamo programi smatraju „potrošačkom robom". ovakav odgovor nas ni ranije nije zadovoljavao, pa nije ni sada. Šta treba da budu takvi programi u Jugoslaviji ako ne potrošačka roba, ili bi možda kod nas trebalo da budu visoka nauka? Valjda nisu namenjeni narodu, nego samo preduzećima, a i ona valjda to trebaju da plate kroz nos.

Drugi deo odgovora je već bio mnogo bolji, naime, planira se zaista izuzetno veliki obim podrške, ne samo po broju mera, već i po zahvatu. Na primer, planira se otvaranje „vruće linije" (hotline), sistem olakšavanja nabavke nadogradnje (upgrade), te stvaranje niza ovlašćenih centara za obuku koje će opet obučiti sam Lotus. Podvlačimo ovo poslednje, jer je to zaista prava novost, a moramo reći, i veoma dobrodošla novost, ako to uspe, i ako takvi centri (koji ne moraju pripadati samom distributeru, već mogu biti i samostalne konsalting kuće) zaista prorade. To bi za nas mogle biti najbolje vesti posle mera Savezne vlade od početka godine, jer je neosporna činjenica da su ovakvi programi dušu dali za planiranje, praćenje i analizu finansijskih tokova preduzeća.

Pitanja je na žalost bilo dosta malo, a poticala su mahom od jednog gosta iz „Interexporta" i vašeg izveštača. Kolega iz „Interexporta" je postavio niz pitanja u vezi nabavke i korišćenja većeg broja programa u okvirima preduzeća, kao i u vezi centralizovane nabavke većeg broja paketa od strane jedne firme kao što je njegova, koja ima poslovnice širom sveta. Gospodin O'Hara je rekao da Lotus ne praktikuje politiku centralnih licenci (tj. kupovine jednog paketa sa doplaćenom dozvolom da se on iskopira tačno definisan broj puta), a što se tiče popusta u vezi kupovine na veće količine, to je stvar lokalnih zastupnika i njihove poslovne politike.



**Neće pomilovanje: Džon O'Hara, direktor prodaje za Evropu**

## PIRATI BEZ AMNESTIJE

Mi smo opet i dalje insistirali na pitanju cena i odnosa korporacije Lotus prema postojećim ilegalnim korisnicima. Konkretno, pitali smo ima li Lotus nameru da omogući postojećim korisnicima da iz stroge ilegale i konspiracije izađu na svetlo dana pomoću tzv. perioda amnestije (grace period), tokom koga bi se za veoma razumne pare ilegalne kopije donosile, uplaćivala bi se neka suma često ne veća od trećine cene programa i dobijala registracija sa pripadajućom dokumentacijom. Gospodin O'Hara nam je odgovorio da Lotus to ne praktikuje, znajući za bogatu istoriju sudskih tuženja ove firme, kao i činjenicu da je *Lotus 1-2-3* bio ama baš poslednji veliki program sa koga je skinuta zaštita. Taj odgovor nas nije ni malo začudio. Naše je mišljenje i u ovom slučaju, kao i prethodno u slučaju korporacije WordPerfect, da je ovo velika greška koja je rezultat nerazumevanja mehanizama našeg tržišta („Jr-žišta"). Argument koji nam je dat je bio da bi takva politika zapravo diskriminisala legalne kupce, koji odmah plate punu sumu, a ispalo bi da neko sa piratskom kopijom dobija isti program za daleko manje pare. Ovo je neosporno tačno, ali na to se može navesti i sledeća činjenica: firma WordStar je izvela jednu takvu akciju u Velikoj Britaniji i postigla čak veoma dobre rezultate pre oko dve godine.

No, moramo reći da je Lotus pokazao veoma visok stepen spremnosti na saradnju, što je dobro i za njih i za nas. Tokom daljeg razgovora, gospodin O'Hara je postavio i ovakvu mogućnost: ponuditi našem tržištu stariju verziju programa sa oznakom 2.01, koja istina nema sve mogućnosti novijih verzija, ali bi se zato mogla ponuditi po „veoma niskim cenama". Ovo bi mogao biti možda čak i najbolji potez ove firme u osvajanju našeg tržišta, jer koliko je nama poznato, upravo ova verzija programa je kod nas najzastupljenija, pa bi legalizacija takvih primeraka prodajom jeftinih originala bila dobar metod otvaranja vrata daljem procesu nadogradnje programa, a ovim bi se sačuvao i integritet kupaca koji su odmah kupili punu verziju za punu cenu. Ako ništa više, ovo je bar jedan pozitivan trend razmišljanja.

Sve u svemu, dobro je što se i Lotus pojavio kod nas na jedan ovakav zvaničan način, posle firmi Borland, Microsoft, WordPerfect, Oracle, Nantuckel, Novell i drugih, evo još jednog igrača iz plejade velikih koji je i došao i doneo lepe vesti sa sobom. Mi možemo samo da poželimo sreću i Lotus-u i njegovim jugoslovenskim zastupnicima.

# U LAŽI SU KRATKE NOGE

**Pre nego što počnete da čitate ovaj članak, verovatno nije loša ideja da pogledate neku (elementarnu) knjigu iz linearne algebre. Nešto (elementarnog) znanja iz teorije izračunljivosti možda je korisno, ali ne i neophodno. Ono što je sigurno potrebno, je doza strpljivosti i želje da se prođe kroz nekoliko matematičkih dokaza. Ako ne verujete da sve to može biti i zanimljivo, onda posebno pažljivo pročitajte članak. Možda će vas on uveriti da dokazi i nisu tako dosadne stvari. Pored toga, naučićete kako da uhvatite nekog u laži. Ili zašto nije najpreporučljivije lagati računardžije. Uostalom, i sama činjenica da niste odustali posle prve dve rečenice ukazuje da vas, ako i nemate možda formalnih znanja, nije lako uplašiti „jakim" rečima.**

Seka R. Javić

Uvek kada odem u banku da podignem ili stavim para osećam se užasno bespomoćno. Nije to što na žiro-računu retko kada nađem neku crkavicu (iako nema sumnje i to doprinosi dubokim osećanjima depresije), već sam proces podizanja ili stavljanja para. Uvek imam utisak, kad recimo podižem pare, da će mi isplatiti na primer samo pola para i mrtvi hladni reći: „Šta čekaš, zar ne vidiš da smo ti dali sve pare". I kako onda, pod milim Bogom, da dokažeš da su ti isplatili samo pola, iako su napisali na kartonče i njihovo terminalče da je isplaćeno sve. Možda bi pomogle suze, možda malo drake i lupanja rukom o sto, a ponekad pomislim da bi se stvar dovela u red ako bi se ponelo neko vatreno oružje. Ali opet, možda bi to samo pogoršalo situaciju, proglase te kako pljačkaš banku, ili da si neurotičan i psihotičan i neotesan.

Kad se sve to lepo uzme u obzir, kao najbolja taktika i ispadne da se uopšte ne buniš. Ili još bolje, da nemaš posla sa bankom. Ali opet, recimo, i kad se izbaci odlazak u banku, ko ti garantuje da račun za struju neće iznositi, recimo, sto puta više nego što si stvarno potrošio struje, ili da neće ispasti da si po dvadeset pet puta dnevno zvao poznanike i tebi ne baš najpoznatije u Australiji, Argentini i Južnoj Africi. E, sad baš da se odrekneš i struje, i vode i telefona ne ide.

Istina, još mi se nije desilo ništa nalik na upravo opisane situacije, ali pitanje poštene transakcije definitivno nije akademsko. Kad banka pošalje drugoj banci stotinak miliona dolara, koje se lepo proknjiže kao sladak osmocifren broj, kako da obe banke budu sigurne (i da mogu da dokažu, ako treba, na sudu) da je stvarno upravo takva poruka poslata s računara na računar. Naravno, može se računati na poštenu bankarsku reč, ali da li postoji alternativa za to beskonačno poverenje?

## RAČUNARSKA KOMPLEKSNOST

Da, kažu najnoviji rezultati teorijskog računarstva. Alternativa se zove interaktivni dokaz i u neku ruku liči na policijsko ispitivanje. U interaktivnom dokazu računar (ili preciznije program koji se izvršava na računaru) se ispituje dokle god se ne pokaže eksplicitno da su od govori međusobno protivurečni ili dok se ne postane uveren da računar govori istinu i samo istinu. Najnoviji teorijski rezultati kažu da su interaktivni dokazi dovoljno jaki da se za svaki problem koji pripada klasi pod imenom PSPACE (P prostor) može proveriti da li je odgovor na njih tačan ili ne.

Ali pre nego što se upustimo u nepregledne dubine teorijskog računarstva, pogledajmo kako stoji uopšte situacija sa proverom rezultata koji su dobijeni pomoću računara. Za mnoge probleme provera je stvarno jednostavna, recimo, ako program invertuje matricu, dovoljno je dobijeni rezultat pomnožiti sa inicijalnom matricom i videti da li je rezultat zaista jedinična matrica, ako sofisticirani program rešava komplikovani sistem jednačina, jednostavno se ubace dobijena rešenja u sistem jednačina i proveri se da li su sada jednakosti ispunjene.

Naravno, postoje slučajevi kada nije sasvim očigledno kako bi se provera izvršila. Ne mislim samo na svoj račun i isplatu u banci, nego, recimo, i na čisto akademske i praktične inženjerske probleme. Recimo, ako vam je dat veliki graf i zadatak je da obojite čvorove sa što manje boja, ali tako da bilo koja dva susedna čvora imaju različite boje, ako računar kaže da je to nemoguće uraditi sa manje od petnaest boja, da li mu verovati ili ne. Ili, data je šahovska pozicija i pitanje da li pobeđuje beli koji je na potezu. Da li biste bili spremni da se kladite u svoj život da je to zaista neminovnost čak i kad vam računar pokaže „pobedničke poteze", ako je sa druge strane šahovske table Kasparov?

Naravno, naivno i prosto govoreći, računar može da proba sve poteze i u vrlo komplikovanoj završnici, ali koliko je realno da će neki računar biti kadar to da učini u nekom razumnom vremenu? Problem leži u efikasnosti izračunavanja. U problemu bojenja grafa, vi možete da uzmete strategiju da probate sve moguće kombinacije kada se čvorovi grafa boje sa četrnaest boja i da vidite da li stvarno postoji neka u kojoj su svi susedi različito obojeni. Ali kada graf ima hiljadu čvorova, takvo probanje i na puno bržim računarima od današnjih znatno nadmašuje milione godina.

Naravno, kad se pokuša da formalizuje šta je efikasan algoritam, a šta nije, dolazi se do izvesne doze kontraverznosti. Međutim, u krugovima teoretskih računardžija postoji konsenzus: efikasan algoritam je algoritam koji zahteva „polinomialno vreme", neefikasan svaki koji zahteva asimptotski duže vreme. „Polinomialno vreme" znači da broj koraka koji se moraju izvršiti u najnepovoljnijem slučaju raste sporije nego neki polinom čiji članovi su stepeni veličine zadatka. Možda smo ovim pokušajem definicije uveli više nepoznatih nego poznatih pojmova, ali par primera će pomoći da se objasni pojam. Recimo, standardan način, koji se uči u ranim razredima osnovne škole, da se pomnože dva broja sa po $n$ cifara zahteva da se učini $O(n^2)$ koraka koji se sastoje od množenja i sabiranja jednocifrenih brojeva. (Naravno, da nas postoje mnogo profinjeniji i brži algoritmi za množenje višecifrenih brojeva, ali o tome drugom prilikom.)

Jednostavna računica pokazuje da ako se prilikom sortiranja koristi algoritam „prvo nađi najveći broj, pa onda počni od početka i nađi sledeći najveći i tako dok se ne sortiraju svi brojevi", traži isto vreme $O(n^2)$. Oznaka $O(n^k)$ znači da je potreban broj koraka ograničen polinomom stepena $k$. I ovog puta postoji brži algoritam ($O(n \log n)$ u najgorem slučaju), ali jasno je da i kad imamo relativno veliki niz brojeva da sortiramo, ovakav algoritam će to urediti u razumnom vremenu. E sad, ako hoćemo da rešimo bojenje grafa gore opisanom metodom grube sile, onda je potrebno učiniti eksponencijalno mnogo koraka.

Teorijske računardžije (ili, kako se oni preciznije i sa više uvažavanja zovu u ovom slučaju teoretičari kompleksnosti izračunavanja) klasifikovali su sve probleme koji se mogu uraditi (jer, naravno, postoje i nerešivi problemi) u vrlo preciznu i komplikovanu shemu, zavisno od toga koliko vremena i računarske memorije problemi zahtevaju. Tri najznačajnije klase su P, NP i PSPACE.

## RAČUNARSKI PROBLEMI SU KLASNO DRUŠTVO

Klasa P sadrži probleme za koje algoritmi koji zahtevaju polinomialno vreme postoje. Kad se takvi problemi proučavaju, cela igra se sastoji u tome da se nađe što je moguće brži algoritam. Nekoliko generalnih principa (podeli pa vladaj, dinamičko programiranje, rekurzija, transformisanje u pogodni domen, „greedy" algoritmi) domišljato primenjenih je najčešće dovoljno da bi se napisao algoritam za problem u klasi P. Posle otprilike pola stoleća rada, opšte je mišljenje da su ključ za najbrže algoritme problema u klasi P odgovarajuće strukture podataka. Biblijom tog dela računarstva smatra se tanka knjižica Roberta Tardjana „Structures and Network Algorithms". Uopšte govoreći, Tardjan, inače dobitnik najcenjenije računarske nagrade Tjuringove nagrade, je koristeći razne zaista sofisticirane (i ponekad vrlo neugodne za implementiranje) strukture podataka, zajedno sa svojim sledbenicima, manje-više isterao brzinu za najpopularnije probleme u klasi P do maksimuma.

Ubedljivo najslavniji problem u ovoj klasi je linearno programiranje, za koje se dugo nije znalo da li se može rešiti u polinomialnom vremenu. Kako linearno programiranje ima vrlo širok spektar upotrebe i relativno ga je lako koristiti, ulagani su veliki napori da se napravi brz algoritam. Interesantno je da je napravljen veliki broj verzija koje su radile sjajno u praksi. Međutim, uvek se za njih mogao konstruisati primer na kome su one zahtevale eksponencijalno vreme. Ali 1979. Leonid Hačijan iz Sovjetske akademije nauka je pokazao da je linearno programiranje problem polinomialne složenosti. Jedini problem sa njegovim algoritmom (ili

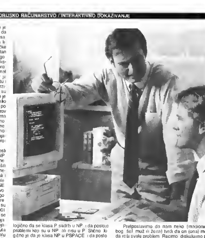
tačnije sa algoritmom koji je on analizirao) je bio taj što je bio užasno spor u praksi. Iako da je bio uglavnom samo od teorijskog interesa. Ali on je startovao bujicu interesovanja za linearno programiranje sa algoritmičke tačke gledanja. I 1984. Karmarkar je lansirao jedan od najkontraverznijih algoritam koji nosi njegovo ime. Da stvar bude gora, za najveće eksperte u programiranju algoritama linearnih programa skoro dve decenije neosporni primat imao je Stenford univerzitet. A Karmarkar je bio napustio studije računarstva na Stenfordu i prešao da doktorira na Kalifornijskom univerzitetu u Berkliju. I stari rivalitet je proradio, ali su se stvari vremenom smirile, pogotovo pošto je Berkli profesor Adler umesto ispita podelio svojim studentima da isprogramiraju deo po deo ceo, pomalo komplikovan, Karmarkarov algoritam. I kada su na svim test problemima Stenford programi bili solidno nadmašeni novim kodom, situacija se smirila. U međuvremenu, računardžije iz celog sveta su, koristeći nove i još više stare ideje, napravili brojne varijacije polinomijalnih algoritama za linearno programiranje.

No, vratimo se teoriji i praksi kompleksnosti izračunavanja. Klasa P se sadrži u klasi NP, gde je NP skraćenica ne za „ne polinomialne algoritme", već za algoritme koji se mogu rešiti u „Ne deterministićkom Polinomialnom" vremenu. Grubo govoreći i izbegavajući važne, ali i komplikovane detalje, klasa NP sadrži probleme koji imaju dva moguća odgovora DA i NE. (Lako je videti da se mnogi problemi lako svode na ovakve DA/NE probleme). Ako je odgovor DA, tada postoji brz (u polinomialnom vremenu) dokaz. Tipični primeri NP problema su faktorizacija broja (da li je broj 1234567001 složen broj), problem bojenja grafa (da li se čvorovi u datom grafu mogu obojiti sa *k* boja ali tako da su susedi obojeni različitim bojama?) i problem nalaženja Hamiltonovog puta kroz graf (da li se može naći put u datom grafu koji prolazi kroz svaki čvor tačno jedanput?). Ako je odgovor na bilo koji od ovih problema DA, tada se odgovor (rešenje) lako i brzo proverava. U slučaju faktorizacije, faktori broja se pomnože i vidi se da li je rezultat upravo broj koji se faktorisao, u slučaju grafa se prebroje korišćene boje i vidi se da li su svi susedi zaista raznobojni, u slučaju Hamiltonovog puta se prebroji da li se kroz svaki čvor zaista prošlo i to samo jednom.

NP klasa ima svog blizanca klasu co-NP. To su problemi za koje se odgovor NE može brzo proveriti. Recimo, ako je pitanje da li je broj 1234567001 prost a odgovor NE, tada se opet samo pomnože faktori. Neki problemi (uključujući celu klasu P kad su problemi formulisani kao DA-NE problemi) pripadaju i klasi NP i klasi co-NP. Rašireno uverenje među teorijskim računardžijama je da klasa P sadrži upravo presek problema koji pripadaju klasama NP i co-NP. (Iako je to uverenje prilično jako, niko ne može ni da ga dokaže ni opovrgne, bar ne za sada).

Konačno smo stigli i do klase PSPACE, koja sadrži probleme kod kojih vreme u kome se mogu rešiti uopšte i nije važno (sem da je konačno kad je problem konačne dužine). Jedino je ograničenje na količinu memorije koja se može koristiti. Česta je analogija sa školskom tablom. Ako se problem može rešiti na školskoj tabli, tako što se ispisuje jedan po jedan kandidat za rešenje, problem je u PSPACE. Naravno, klasa PSPACE je ogromna, ona očigledno sadrži sve do sada opisane probleme, uključujući i šah problem.

Ako ste ikada imali bilo kakav kurs iz matematike, verovatno vas sada najviše interesuje odnos između klasa P, NP i PSPACE. Zvuči



logično da se klasa P sadrži u NP, i da postoje problemi koji su u NP, ali nisu u P. Slično, logično je da je klasa NP u PSPACE, i da postoje problemi koji su u PSPACE, ali ne i u NP. Kažemo logično, jer mnogi problemi koje poznajemo ukazuju da je tako. Ali, bilo kakav dokaz o odnosu ovih klasa ne postoji. Sasvim je moguće (ali kako smo već nekoliko puta rekli i vrlo slabo verovatno), da su ove tri klase u stvari jedna klasa. Redovno se dešava, ponekad su u pitanju vrlo cenjeni naučnici, da poneko dokaže da je P=NP. Ponekad su dokazi izuzetno lepi i domišljati. Poslednji put je profesor sa Univerziteta u engleskom gradu Guephu, pored ostalih, uspeo da ubedi i samog Karpa, koji je definitivno najveći autoritet u oblasti, da je dokaz tačan. Međutim, posle mesec dana Karp je našao rupu u dokazu.

## ZNAM DA NIŠTA NE ZNAM, I DA TI SVE ZNAŠ

Pre nego što se upustimo u opisivanje klase IP, klase problema za koje kratki (polinomialno dugački) dokazi postoje, pogledajmo još jednom šta je „logično" očekivati. Po definiciji, kratki dokazi postoje samo za probleme koji su u preseku klasa NP i co-NP. Znači, u klasi PSPACE postoji zadatak za koji ne postoji kratak dokaz. Da li je to stvarno tako? Konstrukcija kratkog dokaza za takve probleme je nešto tipa kvadratura kruga.

Jeste. Ili možda ipak nije. Tu počinje velika priča interaktivnih dokaza. Upravo kako je kvadraturu kruga moguće rešiti sa proizvoljnom tačnošću, moguće je proveriti za svaki problem u PSPACE-u da li je dati odgovor tačan. Preciznije rečeno, možemo verovatnoću prihvatanja pogrešnog odgovora učiniti malom koliko god hoćemo. I što je najinteresantnije, to možemo učiniti brzo.

Pretpostavimo da nam neko (madioničar, bog, šef, muž ili žena) tvrdi da on (ona) može da reši svaki problem. Recimo, diskutujmo problem izomorfizma grafa.

Data su dva grafa. Pitanje je da li su oni izomorfni, da li oni imaju istu strukturu, tj. da li se ta dva grafa mogu dovesti do poklapanja tako da svaki čvor u prvom ima odgovarajući čvor u drugom, svaka grana u prvom ima odgovarajuću granu u drugom, i da su odgovarajuće grane incidentne odgovarajućim čvorovima u oba grafa. Ovaj problem je NP-kompletan. Ako računar na pitanje da li su dva grafa izomorfna odgovori da nisu, kako se možete uveriti da je odgovor tačan. Uzmete novčić i shodno ishodu bacanja uzmete jedan od grafova, tako da računar ne zna koji ste uzeli. Zatim, taj graf lepo precrtate proizvoljno i drugačije označite čvorove. I pokažete novonastali graf računaru i pitate koji originalnom grafu je novonastali izomorfan. Ako računar stvarno može da reši problem, on će lako odgovoriti. Ako ne može, on ima 50% šansi da pogreši. Naravno, računar može da ima sreće i slučajno pogodi odgovor. Ali kada se navedena procedura ponovi par stotina puta (ili više), verovatnoća da računar stalno pogađa tačan odgovor postane vrlo brzo zanemarljivo mala.

Interaktivni dokazi su prvi put razmatrani 1985. godine. Shafi Goldwasser, Silvio Micali i Charles Rackoff su ih primenili u kriptografiji. Pošto su protokoli koje su oni koristili bili potpuno specifični za diskutovani problem, zavladalo je mišljenje da interaktivni dokazi postoje samo za vrlo ograničene klase problema. Niko nije ni pomišljao da interaktivni dokazi postoje za sve probleme u PSPACE.

Ali negde počevši od prošle zime, bujica rezultata je dokazala upravo to. Pogledajmo samo najvažnije (ili bolje rečeno) najinteresantnije

je. Ono što je najinteresantnije nije samo činjenica da je dokaz doneo potpuno neočekivan rezultat, već i to da je izuzetno jednostavan.

Ključ rešenja je bio u takozvanom izračunavanju permanenta. Permenent je blizak rođak determinante: oboje se definišu rekurzivno na kvadratnim matricama. Za one koji znaju kako se izračunavaju determinante, lako je definisati permanent. Pri izračunavanju determinante sve minus znakove zamenili plusevima i rezultat je permanent. Izbacivanje svih minusa iz izračunavanja permanenta čini suštinsku razliku sa tačke gledišta kompleksnosti izračunavanja: dok je determinantu lako izračunati u polinomijalnom vremenu, permanent je „kompletan" problem u klasi #P koja leži između klasa NP i PSPACE.

Kompletni problemi su jedno od najvećih blaga teorijskog računarstva, i počevši od Kuka i Karpa, mnoge računardžije su stekle svetsku slavu otkrivajući takve probleme. Poenta je u sledećem: ako je problem kompletan za neku klasu, to znači da se svi problemi u toj klasi mogu u polinomijalnom vremenu transformisati na dati problem i stoga rešiti u vremenu iste dužine kao i dati kompletan problem. Ubedljivo najpopularniji kompletni problemi su NP kompletni problemi koji uključuju mnoge poznate kombinatorne probleme, kao što su Hamiltonov put i bojenje grafa. Upravo činjenica da niko nije uspeo da razreši nijedan od NP-kompletnih problema u polinomijalnom vremenu stvara jako ubeđenje da to nije ni moguće uraditi. (Spisak poznatih, popularnih i „često sretanih" NP kompletnih problema je dug nekoliko hiljada problema.)

Činjenica da je izračunavanje permanente u #P čini razumnom pretpostavku da je izračunati permanent teže nego determinantu, zato što bi svaki algoritam za izračunavanje permanenta, automatski razrešavao sve ostale probleme u #P, koji bi trebalo da budu najteži problemi u NP.

Radeći na problemima za testiranje programa, Richard Lipton sa Princeton univerziteta je pokazao da se program koji izračunava permanent može „randomizirati" tako da radi tačno „skoro uvek". „Randomizirati" znači ovde korišćenje generatora brojeva koji statistički čine uniformnu raspodelu, a „skoro uvek" znači da se verovatnoća greške može učiniti proizvoljno malom. Noam Nisan sa MIT je koristeći Liptonov rezultat dokazao da se izračunavanje permanenta može proveriti koristeći višestruke interaktivne dokaze — specijalnu varijantu interaktivnih dokaza gde se istovremeno ispituju dva „optužena" posebno, tako da jedan ne zna za odgovore drugog. (Recimo, ispitujete suprugu i njegovu potencijalnu ljubavnicu kako su proveli juče poslepodne i poredite njihove odgovore.)

Koristeći e-mail, Noam je obavestio svog druga sa studija na Berkliju Karlofa koji je u međuvremenu dobio posao na Čikaškom univerzitetu. Karlof i njegova grupa (Carsten Lund i Lance Fortnow) su kroz tri nedelje vratili e-mail koji je pokazivao da višestruka ispitivanja uopšte i nisu potrebna: dovoljan je i interaktivan dokaz. (Znači, manje-više, bolje je koncentrisati snagu na ispitivanje muža, on će i onako sve sam priznati.) Glavna ideja, ne u ispitivanju muža, već u izračunavanju permanenta, je da se program koji je izračunavao permanent pita redom da izračuna sve manje i manje matrice koje se sadrže u većoj i tako forsira da laže o njihovim permanentima, ako je lagao o originalnom. Na kraju program mora lagati o permanentu matrice koja ima samo jedan broj i koji je upravo taj broj.

Pogledajmo sve to malo detaljnije. Pretpostavimo da proveramo rezultat izračunavanja matrice $M=(m_{ij})$. Prvi korak je da se program pita za permanent svih minora prve vrste. (Minori određene vrste matrice su matrice koje nastaju kada se ta vrsta i proizvoljna kolona jednostavno izbace iz originalne matrice.) Pošto je $per\ M = m_{11}\ per\ M_1 + \dots + m_{1n}\ per\ M_n$ (koristeći očigledne oznake), lako je proveriti da li su rezultati konzistentni. Naravno, sada je problem u tome, kako zaključiti koji minor je uzrok pogrešnog rezultata i njega dalje razlagati da bi se ispoljila protivurečnost.

Pobednički trik je u tome da se svi minori kombinuju u jedan, uzimajući po dva istovremeno. Jedino znanje matematike je osnovna teorema linearne algebre: da polinom stepena $n$ ima najviše $n$ nula. U razmatranom slučaju polinom se konstruiše koristeći permanent matrice $(1-x)A + xB$ gde su $A$ i $B$ minori. Po definiciji permanenta, ovaj polinom ima vodeći član stepena ne većeg od $n-1$. Ako pitamo program za koeficijente $per(1-x)A+xB$, $n-2$ puta, tada ćemo pored nula polinoma za $x=1$ i $x=0$ saznati sve nule. Poenta je u tome da ako je program dao pogrešne vrednosti za $A$ i $B$, polinom je takođe pogrešan. A to opet znači da ako izaberemo slučajno tačku $x$, vrednost polinoma će u toj tački skoro sigurno biti pogrešna, jer dva različita polinoma mogu da imaju iste vrednosti u najviše $n-1$ tački.

Dalji dokaz je lako dovršiti koristeći indukciju. Međutim, ako imate problema sa shvatanjem dokaza, a ipak biste želeli da ga shvatite do kraja, verovatno dobra ideja je konsultovati bilo koji kvalitetan i standardan udžbenik linearne algebre.

## IP=PSPACE

Znači, zaključak je Čikago grupe: IP je u najmanju ruku veliki koliko i #P. S druge strane, tu je bio i poznati rezultat da se IP sadrži i u PSPACE. Naravno, svi upućeni sada su znali da je put do slave dokazati da je IP=PSPACE. Najbrži je bio Shamir iz Weizmann Instituta u Izraelu. (To je onaj isti Shamir koji je napravio i prvi RSA kriptografski protokol sa javnim ključem. Ta priča ovih dana doživljava drugu mladost, ali i o tome drugom prilikom.)

Shamir je koristio u dokazu PSPACE kompletan problem, poznat pod oznakom QBF. QBF je u stvari kvantifikovana Bulova formula u kojoj logičke jednačine imaju i kvantifikatore „za sve vrednosti" i „postoji vrednost za koju". Naravno, pitanje je da li je data QBF tačna ili lažna. Shamir je koristeći analogiju sa protokolom kojim je dokazano da je izračunavanje permanenta u IP, konvertovao QBF u niz izračunavanja polinoma više promenljivih. Glavni problem je bio onemogućiti tendenciju brzog rasta stepena polinoma, pri svođenju na sastavne probleme, ali tu je Shamir uposlio novi trik: uvođenje novih veštačkih promenljivih. Time je priča dobila neočekivan, ali jasan kraj: IP=PSPACE. Interesantno je da od kada je Nisan poslao e-majlom svoj prvi rezultat do Shamirovog dokaza da je IP=PSPACE prošlo manje od mesec dana. Postalo je jasno da su dve najveće opasnosti za teorijske računardžije: ne biti priključen na elektronsku poštu i ići na skijanje.

Ako vam sve ovo izgleda interesantno, ali nerazumljivo, onda je verovatno najbolje da pročitate knjigu od M. Garey i D. Johnsona „Computers and Intractability" i da pročitate članak još par puta. Ako vam sve izgleda jasno, ali kao da je to tek početak interesantnih rezultata, tada vam preporučujemo da se za 3 dolara godišnje pretplatite na ACM SIGACT koji redovno donosi izveštaj o najnovijim trendovima u teorijskom računarstvu u (relativno) lako shvatljivom obliku.

## DA LI POSTOJI BOG



*Bio jednom jedan matematičar, pametan i lep, koji je bio bolji od svih ostalih. I kao svoje najveće otkriće, mnogo bolje od svih ostalih, jednog dana on pronađe jednačinu za boga. Naravno, bila je to vrlo, vrlo komplikovana jednačina sa mnogo, mnogo nepoznatih. Nije ni bilo govora da se ona reši na papiru ili kako to matematičari kažu analitički. Ali, kako je to bila era računara, trebalo je samo naći dovoljno veliki računar koji će imati (1) dovoljno veliku memoriju da smesti sve promenljive, i (2) biti dovoljno brz da obradi ogromnu količinu informacije. Jednom, kad on završi posao, konačno će sve naučne tajne biti razrešene i stanje večne sreće i savršenstva će zavladati.*

*I projekat se toliko svideo ljudima u vladi da su oni bez trunke oklevanja rešili da sve računare u glavnom gradu, svaki mikroprocesor, kompajler i linker i sve ostalo potrebno i ne baš sasvim potrebno stave zajedno u mrežu i pitaju novi, ogromni računar: „Da li postoji bog?"*

*Na žalost, ili možda na sreću, program koji je rešavao jednačinu prvo je trebalo da proceni koliko vremena će trebati da se reši jednačina. I odgovor je glasio: najmanje per stotina godina, a možda i više.*

*Naravno, ljudi iz vlade su mislili da to nije dovoljno brzo jer, zaboga, uskoro će izbori, a oni nemaju čime da impresioniraju birače. A ako bi imali rešenje za sve probleme, ko ne bi glasao za njih?*

*I tako oni odlučiše da apsolutno sve računare u zemlji i svu ostalu prateću računarsku skalameriju stave u jednu mrežu i da novi VELIKI računar upitaju: „DA LI POSTOJI BOG?"*

*Ovog puta računar je dao znatno optimističkiju procenu potrebnog vremena. Tačan odgovor će se znati za desetak godina. Nastupilo je opšte uzbuđenje. Naravno, ljudi iz vlade smatrali su da to nije ni izbliza dovoljno brzo. Zaboga, izbori samo što nisu počeli.*

*Ali kao i uvek vlada je imala rešenje. Postala je svoje izaslanike u celi svet. I svim vladama sveta objasnila svoj sjajan projekat i neverovatne koristi koje iz njega proističu. I posle izuzetnih diplomatskih i političkih akcija, sve vlade svih zemalja su rešile da sve ostale projekte treba otkazati i raditi se samo jedan, onaj pravi. Naravno, bilo je tu problema oko toga kome će pripasti koji deo slave (i ne samo slave), ali kako su svi bili impresionirani, sve je konačno ugovoreno. I svi računari na planeti Zemlji su stavljeni kao delovi jednog velikog, neverovatno velikog računara. I svi softverski problemi kompatibilnosti su rešeni. I novi V E L I K I računar je startovan. I postavljeno mu je pitanje: „D A L I P O S T O J I B O G ?"*

*I računar je odgovorio:*

*„DA, SADA POSTOJI!"*

*Seka Bajt*

**Operativni sistemi**

## NOVI DRAJVERI ZA WINDOWS 3.0

Microsoft uskoro počinje sa prodajom *Windows Suplemental Driver Library*, setom prinlerskih, ekranskih i drajvera za ostale periferije (device drajveri) za *Windows 3.0* koje su razvili proizvođači hardvara, a Microsoft sertifikovao. SDL, kako se paket skraćeno naziva, uključuje i drajvere za HP LaserJet II i III za koje u Microsoft-u kažu da daju najbolji otisak iz *Windows*-a. Prva verzija SDL-a takođe sadrži drajvere za AMT, Bitstream, Brother, Canon, IBM, Kodak, Okidata, Olivetti, Seiko i Star štampače.

Ekranski drajveri, uključujući i Super VGA, su razvili ATI, Chips & Technologies, Compaq, Graphic Software Systems, Video 7 i Western Digital, a drajver za mreže je, naravno, razvijen u Novell-u.

SDL se može naručiti direktno od Microsoft-a na (američki) telefon (800) 426 9400 po ceni od $20.

**Tvrdi diskovi**

## 535 MB NA 3.5 INČA

Maxtor je najavio novi 3.5" hard disk sa rekordnim kapacitetom za ovaj format. LXT-535, kako je nazvan, može da smesti 535 MB uz prosečno vreme pristupa od 12 ms, a opremljen je SCSI ili AT interfejsom i cenom od $1450. Uređaj je u fazi testiranja i treba da se pojavi na tržištu početkom sledeće godine. Onima koji ne mogu toliko da čekaju nudi se „prvi-do-njega" model LXT-437 kapaciteta 437 MB po ceni od $1250.

**Računari**

## IBM OBARA CENE ZA PS/2

IBM je rešio da obori cenu modela PS/2 80-121 za 7%. Mašina se sada prodaje za $6945 umesto ranijih $7495. Cena obuhvata 20 MHz 386 sistem sa 2 MB RAM-a, 120 MB hard disk i 1.44 MB flopi disk. Veliki Plavi je najavio i sniženje cena cele PS/2 serije, a modeli 30-001, 30-021, 80-111 i 80-311 se od 27. decembra više neće prodavati.

**Portabli**

## OBOJENI T5200

Toshiba je iznela na tržište dve nove konfiguracije svog 386 portabla T5200 – jedan od novih modela raspolaže kolor LCD ekranom. Toshiba se tako pridružuje NEC-u i Sharp-u u proizvodnji portabl modela koji koriste kolor LCD ekran sa pasivnom matricom, što predstavlja tehnologiju koja se smatra međukorakom do ekrana sa aktivnom matricom.

Toshiba T5200C koristi 20-MHz 386 procesor, ima 200-MB hard disk, koristi mrežni napon (AC) i teži 7.65 kg. Veličina i potrošnja kolor LCD ekrana za sada ne dozvoljavaju baterijsko napajanje.

Cena od $9499 svakako nije mala, ali je ipak za „samo" $1000 veća od cene istog modela sa gas-plazma ekranom (poređenja radi, kolor LCD sa aktivnom matricom firme Dolch & Dauphin košta $4000 više od monohrom modela).

T5200C tako ulazi na vrlo nesigurno tržište na kome se već žestoko bore Sharp Multi Color 386 i NEC ProSpeed CSX. Prednosti T5200C u odnosu na konkurente su, pre svega, niža cena i manja težina, i naročito značajno kvalitetnija slika sa boljim i gušćim bojama, višim kontrastom i punom VGA rezolucijom od 640 x 480. Poboljšanje je postignuto novim rešenjem filtera i boljim pozadinskim osvetljenjem. Dodavanje boja na ekrane sa pasivnom matricom zahteva dodavanja crvenog, zelenog i plavog filtra preko odgovarajućih dioda, čime se smanjuje intenzitet pozadinskog osvetljenja. Toshiba koristi STN (super twist nematic) monohrom displej i filtere sa dispergovanim pigmentima, a za pozadinsko osvetljenje 4 fluorescentne lampe sa hladnom katodom.

Vlasnici starog T5200 će moći, počev od prvog kvartala iduće godine, da zamene plazma ekran za kolor LCD po ceni koja još nije utvrđena.

Kontakt adresa: Toshiba America Information Systems, Computer Systems Division, 9740 Irvine Blvd., Irvine, CA 92718, USA, tel. (714) 583-3000.

**Stono izdavaštvo**

## QUARKXPRESS POD OS/2 I RISC

QuarkXpress, jedan od vodećih DTP paketa za „mekintoš" bi tokom sledeće godine trebalo da se pojavi na tržištu i u verzijama za OS/2 i IBM RISC System/6000. Prvobitno je razmatrana i *Windows* verzija, ali su u firmi došli do zaključka da su mogućnosti povezivanja i rada u mreži koje *Windows* nudi suviše slabi za mogućnosti *Xpress*-a, tako da se za sada ostalo na OS/2 i RISC verzijama. Ovaj potez treba posmatrati u svetlu sve žešće konkurencije na DTP tržištu – *Quark*-ov glavni takmac, Aldus, već izvesno vreme pored verzije za „mekintoš" prodaje i *Windows* i OS/2 verzije svog *PageMaker*-a, dok je *Ventura* upravo izašla i u „mekintoš" verziji.

Kontakt adresa: Quark, 300 South Jackson St., Suite 100, Denver, CO 80209, USA, tel. (303) 934 2211.

**Portabli**

## COMMODORE LAPTOP

I CBM se najzad pojavio sa laptop modelom – mašina je teška 2.6 kg, koristi 12.5-MHz 80C286 procesor, prima do 5 MB RAM-a, ima VGA ekran sa pozadinskim osvetljenjem, tastaturu sa 82 tastera, 20 MB hard disk i 1.44-MB flopi disk, a trajanje baterija iznosi dva sata. Od „viala u svet", tu su serijski i paralelni port i slot za modem. Uz mašinu se isporučuje MS DOS 4.01. Cena: $2999.

Kontakt adresa: Commodore Business Machines Ltd, 1200 Wilson Dr, West Chester, PA 19380, USA, tel. (215) 431 9100, fax (215) 431 9156.

**Skeneri**

## SCANMAN 256

ScanMan 256 je novi Logitech-ov ručni skener sa 256 nijansi sivog i rezolucijom od 400 dpi. Po konstrukciji je sličan prethodnim Logitech-ovim skenerima, ali uz nekoliko funkcionalnih poboljšanja. Dizajniran je za rad pod *Windows 3.0* i opremljen softverskim paketom nazvanim *Ansel* koji omogućava lako i brzo skenovanje i editovanje direktno iz *Windows*. Pored toga, uz skener se dobija i *DOS Scan* utility koji omogućava skenovanje, pregledanje i snimanje slika na disk iz DOS-a, bez potreće za ulazak u *Windows* ili *Ansel*. Za razliku od *Ansela*, *DOS Scan* ne raspolaže mogućnostima za editovanje.

Podešavanje skenera se obavlja mikroprekidačima na telu skenera, a prateći softver automatski očitava postavljenu konfiguraciju i prilagođava joj se. Iako je podešavanje vrlo lako, problemi nastaju kod izbora kombinacije rezolucija-broj nijansi sivog: obična sličica veličine 5 x 5 cm pri rezoluciji od 300 dpi i 256 nijansi sivog zahteva skoro 2 megabajta memorije. Sve u svemu, reč je o fleksibilnom i moćnom malom uređaju koji može biti vrlo koristan za skenovanje malih dokumenata ili slika, pod uslovom da korisnik raspolaže sistemom dovoljnih memorijskih mogućnosti.

Cena: $499.

Kontakt adresa: Logitech Inc., 6505 Kaiser Dr., Fremont, CA 94555, USA, tel. (415) 795-8500.

**Stono izdavaštvo**

## AKTIVNI DOKUMENTI

Interleaf je najavio novu seriju programa za DTP sa implementiranom *active document* tehnologijom. Aktivni dokumenti predstavljaju elektronska dokumente koji prema rečima predstavnika firme „mogu da pristupe, procene i deluju prema informacijama". Reč je o novoj Interleaf 5 seriji koju čine 7 programa – Interleaf Engineer, Professional Writer, Illustrator, Production, Academic i Passport.

Svi programi se odlikuju reprogramabilnom strukturom, što omogućava lako i efikasno prilagođavanje specifičnim potrebama i unifikaciju na nivou grupe korisnika. Programerima stoje na raspolaganju alati kojima se mogu kreirati dodaci glavnim programima, modifikovati program ili napraviti od postojećeg programa „potpuno novu aplikaciju".

Interleaf 5 u verziji za UNIX radne stanice će se pojaviti na tržištu u decembru, DOS i VMS verzije se očekuju na proleće, a Mac verzija na leto sledeće godine. Što se cena tiče, najjeftiniji je Passport koji će koštati oko $2500, dok će cene ostalih programa ići i do $15.000. Cene za PC i Mac verzije još nisu poznate.

Kontakt adresa: Interleaf, Ten Canal Park, Cambridge, MA 02141, USA, tel. (617) 577-9800.

**Lokalne mreže**

## LAN NA 220 V!

Povezivanje računara preko obične električne mreže kakvu svi imamo u stanovima i na radnim mestima? Moguće je. *Verran Datalink & DPS* (Dedicated Printer Sharer) sistem za povezivanje računara i deljenje printera omogućava povezivanje mikro i mini računara sa drugim PC-ima, terminalima ili printerima preko postojeće električne instalacije i zidnih utikača. Prenos je siguran i tačan zahvaljujući solidnim tehnikama za enkripciju i korekciju grešaka, i odvija se brzinom od 16 Kbps. Datalink radi sa PC kompatibilcima i „mekintošima".

Ovakva vrsta prenosa ima tri prednosti u odnosu na klasične mreže slične namene: nema softverskih konflikata niti problema sa RAM-om zato što nije potreban nikakav softver, veza je brza, jeftina i laka za postavljanje budući da instalacija već postoji. I, na kraju, moguće su sve kombinacije opreme, pošto se veza ostvaruje preko standardnih serijskih i paralelnih portova. Cena: $395.

Kontakt adresa: GEC-Marconi Software Systems, 12110 Sunset Hills Rd, Suite 450, Reston, VA 22090, USA, tel. (703) 648 1551, fax (703) 476-8035.

# SAD I NIKAD VIŠE

**Vec duže vrijeme u računarskom svijetu figurira jedna stara istina – pionirski dani računarstva su zauvijek prošli; umjesto zanesenjaka koji su pravili mašine za svoju zabavu i dobrobit oatalih, nastupilo je vrijeme bezobličnih lica u trodijelnim odijelima po čijem nalogu se prave računari koji sa epitetom revolucionaran imaju zajedničko samo početno slovo (da li vam imena C 16, C 166 i C +4 išta govore?). Ovogodišnji Personal computing '90 (nasljednik negdašnjeg PCW-a) ostavljao je utisak jednog (u poređenju sa negdašnjim PCW-om) VEOMA DEPRESIVNOG MJESTA!!!**

## NEKAD I SAD

Prije 12 godina Anđelko Zgorelec – osnivač časopisa PCW (personal computer world) osnovao je istoimeni sajam – Personal Computer World Show (PCW Show) Sajam se pokazao kao pun pogodak. Iz holova jednog londonskog hotela selio se svake godine u sve veću i veću dvoranu Dugo se zadržao u hali Olimpia sve dok i nju nije prerastao i preselio se u najveću londonsku izlozbenu halu – Earl's Court Exibition Hall Sajam je jako rastao sve do 1988 Tada je vašem izvještaču za obilazak sajma jedva dostalo 5 dana po 9 sati! A onda je 1989 promjenjena koncepcija sajma Čuveni PCW Show je razdvojen na dva sajma je dan posvecen igrama a drugi striktno poslovni I onda je nastupio antiklimaks Igrački sajam je nastavio da sjajno radi dok je poslovni dio morao da sve počne ispočetka Ukratko PCW Show je izgubio svoju draz koju su mu davali bezbrojni „ludaci" zajedno sa svojim još ludim proizvodima kao i gomila malih proizvođača sa ponudom svega i svacega

A sad je sve drugačije Vaš izvještač je nezvanično saznao (iz dobro obaviještenih krugova) da je ovogodišnji sajam (upravo zbog totalno promjenjene koncepcije) na finansijskom gubitku od oko 1 milion funti! Jer stvar je jednostavna – ovo što se zbiva nema više nikakve veze sa PCW om i zapravo je novi sajam (kralj je mrtav – živio kralj) Zato su organizatori da bi privukli one velike ponudili prostor BESPLATNO! I ne samo to Oni su im još i PLATILI!!! Pa sad neka neko kaže sa nije lijepo biti IBM ili APPLE Ali nemojte na osnovu toga misliti da je ovogodišnji sajam PERSONAL COMPUTING '90 bio sasvim neinteresantan – naprotiv Ono što se imalo vidjeti vrijedilo je dolaska iz Jugoslavije A oni koji su došli bili su i veliki i mali su sta da pokazu

## SVE SVOJE SOBOM NOSI

Od velikih niste mogli da ne zapazite AMSTRADA koji je kao i obično zauzeo prostor po principu sve svoje (kancelarije) nosim sa sobom (na sajam) i tamo radim business as usual A nakon Amstrada tu su se smjestili i Apple Autodesk Bitstream Corel systems corporation D Link Commodore, Fox software HP Hitachi (ovdje sada slijede tri slova velikog plavog koje nećemo objavljivati radi uštede prostora), Locomotive software Mc Grow-Hill Microsoft NeXT computers Olivetti Panasonic Precision software Roland Star Micronics Tandon Taxan Travelling software Tulip Wordperfect, kao i Ashton Tate (na štandu NeXT a) ali o njemu kasnije Kao što ste mogli primjetiti na sajmu nije bilo Borlanda EPSONA i još pokoje firme i firmice Ali to je izgleda pravilo (npr ili Microsoft ili Borland i slično)

## Saša Svitlica

Amstrad je na svom štandu predstavio novu seriju kompjutera nazvanu Generacija 3 Katalog im je fino dizajniran, što se na bi moglo reći za računare dizajn im je „srednja žalost" a o cijenama sudite sami PC3386SX (386SX 1MB 40MB – 28ms 3 5" flopi HD VGA card) i monohrom 12" 1399GBP sa 14" colour (valjda misle 640 * 400) 1499GBP a sa 14" hires (800 * 600) i 64K keša – 1599GBP dok model ispod PC3286 (80286 1MB 40 MB hard i VGA card) sa istim modelima monitora respektivno košta 999 1099 i 1199GBP Iz ove generacije ostao je još jedan XT koji zavisno od konfiguracije flopija 1 ili dva 3,5 ili 5 25" i hard diska 30MB (i ovdje je VGA standard) košta između 549 i 999 funti! Što je previše – previše je pa makar i od Amstrada za te pare 999GBP = 3000DEM = 21000 YUD mozete kupiti štošta opreme i kod nas a ne nekog XT a pa makar bio i sa VGA kartom i 14" kolor monitorom Inače Amstrad i dalje isporučuje seriju PC 1512 i PC 1640 kao i PC 2000 te PCW seriju koji se drže na starim pozicijama Zauzvrat više nema ni pomena PPC ovima ni Spectrumima ali su zato tu telefaxi i to 3 modela sa cijenama 399 499 i 699 GBP (sve Amstradove cijene su bez VAT-a – poreza od 15%)

## OD ŠTANDA DO ŠTANDA

Na sljedećem štandu vidio sam nešto lijepo i korisno na starom polju obnavljanja traka za matrične štampače firma OFFICE 21 predstavila je SPREJ RE-INK za obnavljanje istrošene trake Cijena – sitnica 12 95 GBP sa uključenim VAT om Upotreba je krajnje jednostavna otvorite traku sa gornje strane i u smotuljak ušpricate dvije čajne (ah ti Englezi) kašike tekućine (kako mjeriti sprejanu tekućinu kašikicama nije objašnjeno) i pustite nekoliko minuta da se stvar osuši i traka je spremna za ponovnu upotrebu Sajamski prostori su posebno pogodni za izvođenje čarolija najrazličitijih vrsta koje onda jadnim (čitaj prevarenim) kupcima nikako ne polaze za rukom u kućnom ili radnom okruzenju Jedan sprej je dovoljan za 30 ribona i to je to Umalo da zaboravim sprej ne sadrži CFC (ozone friendly) Šta čekate svi u kupovinu! Kad smo već kod obnavljanja evo i jedne firme koja nudi obnavljanje tonera E to je već zeznulo Iz ličnog iskustva znam kako rade ponovno punjeni toneri (o tome vjerovatno uskoro u nekom drugom članku) i došao sam prilično ljubopitiv Firma je LAZERFILL a produkti obnovljeni HP toneri I ne samo obnovljeni već i popravljeni Naime firma kupuje stare (istrošene uništene smrvljene zgažene spljosnute ) HP tonere za 5 GBP/komad A prodaje ih ugodno dotjerane i popunjene sa 30% tonera više za pukih 40 GBP (može ići i do 30 ako nabavljate veće količine) Tel 99 44 71 3597679 Fax 9944 71 359 7761

Evo sada i jedne opake softverske firme – Precision software – tvorca programa Superbase ovog puta verzije 4 0 i to za Windows okruženje Po mom mišljenju ljudi više vole ljubazniji (i jednostavniji za upotrebu) program nego mogućnost spremanja slike i zvuka u bazu podataka Ali šta je tu je – ljudi furaju svoj fazon, a sudeći po količini propagandnog materijala i količini besplatno dijeljenih disketa sa demo verzijama ovog programa – ne ide im baš nejodlje Uz besplatne diskove dobijete i „knjižicu" – sitnica samo 92 strane sitnog teksta – upotstvo za rad sa demo verzijom! A kao stvar je jednostavna za upotrebu – radi pod Windowsima He

Medjutim tek kad sam bio propisno natovaren Superbase katalogom ugledao sam štand jedne firme sa neupadljivim imenom – VIDEK koja me je počastila katalogom A4 formata sa 66 stranica koje nude SVE MOGUĆE VRSTE KABLOVA I KONEKTORA Pošto sam ja jedno ubogo ljudsko stvorenje koje dolazi iz jedne drzave u kojoj poziva na odbranu „nacionalne ugrozenosti" ima na svakom koraku a najobičniji telefonski utikač ne možete naći ni za lijeka stvar je bila jasna A u katalogu nema čega nema paralelnih i serijskih kablova za mali milion računara (od kojih su neki čista egzotika – npr Einstein) zatim kablovi koji glume modeme između različitih računara i to u svim mogućim kombinacijama zatim MIDI kablovi mreže pasivni i aktivni elementi mreža pa sva sila konektora printer šerera i bafera video kablova i audio kablova Evo za kraj samo jednog podatka obični paralelni kabl 1 8 metara za PC e košta 100 din a u količinama od 50 komada pa naviše 50 din! Nakon ovoga ja sam bio siguran samo u jedno ovi ljudi imaju svega osim nacionalno osviještenih sluzbenika i to je naša izvozna šansa za još jeden posao stoljeća! Tel 9944 81 204 6690

Zsoft je ponudio PC Paintbrush – kastomibilni meniji boja kontrola skenera puno boja i ništa više Fax 991 404 427 1150

A sajmom je dominirao – ARCHE Firma za koju vjerovatno niste ni čuli a stvari koje nudi nisu ništa posebno – prosječno dizajnirani računari, prosječnih brzina i prosječne cijene, ALI vanprosječna podrška i održavanje To ukratko znači da svaki kupac dobije 2 godine garanciju i uz to serviser dolazi kad god to kupac zaželi Hajde sada da vidimo ko to kod nas nudi iole nešto slično Nije sramota biti šraficigeraš – to Arche jeste ali je i te kakva sramota biti neozbiljan i ne servisirati mašine koje prodaš ili što je još gore ne isporučivati uplaćeno A lista takvih YU firmi bi povećala obim fajlove na SEZAMU bar duplo Ali da se mi naslavimo praviti Englezi i obilaziti sajam

Naravno tu je bio i Commodore A tamo je bilo mnogo novih – spotova Da izgleda da bi



Commodore uskoro mogao nadmašiti Pepsi i Koka-kolu po količini novih reklama za iste stvari koje to išlo ne mogu još raditi. Tako se i ovog puta mogla vidjeti izuzetno urađena reklama koje bi se mogla nazvati „Kako čike iz Komodorovog rukovodstva zamišljaju MULTIMEDIU i amigu u vezi s tim." Od hardvera je pokazana AMIGA 3000. Te X window programer s toolkit verzija 4.0 u proizvodnji GfxBase, čiji se prospekt dijelio u fotokopiranoj formi. Izgleda da je stvar završena u zadnji tren (mislim i na prospekt i na softver). Tu je bio još i AMIGA VISION – navodno program za MULTIMEDISANJE na AMIGI. Elem, ništa što prije nije viđeno. Hvala dragom bogu da su komodorovci konačno shvatili šta AMIGA najbolje može – grafika i zvuk i da su se na to bacili, ali avaj opet ne svom snagom, već nekako mlitavo sve u stilu „Hajde da probamo, pa ako uspije dobro je, a ako ne, nema veze".

Za razliku od Komodora, Worperfect je znao da ima dobar proizvod kog treba gurati i to je upravo uradio. Wordperfect ćete od sada nalaziti i na SCO XENIX-u, kao i na NeXT-u! Bravo!

A na štandu 1127 dijelila se brošura na kojoj je pisalo „Zašto će stotine ljudi zastajati na našem štandu?" Moj odgovor glasi: „Da bi vidjeli jednu od većih SKUPIH gluposti na koju možete spiskati svoje pare." Riječ je o kartici koja je ujedno genlock i prikazuje sliku u jednom prozoru pod Windowsima, naravno u VGA. Sve to za pukih 299GBP. Znatiželjno sam upitao momka koji je to konstruisao čemu to sve služi, a odgovor je bio jednako produhovljen: „Nemam pojma!" Čovjeku ne treba zamjeriti – bar je bio iskren.

Treba li vam sto za računalo? Ako treba evo vam i cijene: 139 do 179 GBP (2800 – 3200 YUD). Radi se o praktičnim stolovima na točkićima sa više polica na koje slažete opremu. Tel 9944 993 881912.

Od proizvođača printera jako je ugodno iznenadio Panasonic. Prije svega tu su bili veoma robusni matrični štampači, ali i laserski štampač KX P4455 sa čistim Postscriptom i 11 strana u minuti. Cifra je 2,7 kilofunti. Ipak ono što je bilo najnovije kod Panasonice su bili optički izbrisivi diskovi, ali bez magnetne komponente! Naime to su čisti optički diskovi gdje laser radi sve – piše, briše i čita. Tačnu cijenu nisam mogao saznati ali će biti oko 5000 USD.

A onda je tu bio IBM. Pored njihovog štanda nije uputno prolaziti triško najeden. Stvarno, taj izgled novog PS/1 kojim IBM ponovno pokušava prodrijeti tamo gdje neslavno propali PC jr. nije mogao – u domove ljudi širom svijeta. Ja bih svome djetetu prije kupio lutku sa likom gremlina, djete bi se manje plašilo noću. Dakako računar je RUUUUUUUŽAN. Ali kome oni to misle prodati kilavog AT-a na 10 MHz sa 512K RAM-a. Pa makar i sa sloganom „Za pet minuta vi ste u biznisu." Kakvom biznisu? Prodaje ovog ružnog pačeta, možda?

Ali zato HP nema tih problema. Hit ove godine predstavlja HP III, a slijedi i verzija D. Ovo se odnosi na duplu kasetu za papir i mogućnost dvostrane štampe. Njegova cijena je 3000 GBP. Osim ovoga HP je najavio i nove niže cijene za svoje plotere i to HP Color pro (sa 895 na 695), HP 7475 (sa 1295 na 995), HP7550 plus (sa 3595 na 3495). Za YU tržišta je naročito bitno sniženje elektrostatskih plotera i to modela 250 (sa 21950 na 19950) i modela 225 sa 24950 na 22950 GBP). Osim ovoga HP je najavio i novi deskjet printer – HP DeskJet 500 sa cijenom od 599 GBP. Ovaj model je došao umjesto modela HP DeskJet i HP DeskJet PLUS koji su koštali 769 i 649 GBP respektivno. Od softvera je predstavljen novi Scan Gallery – program za podršku ScanJet PLUS skenera i to je verzija ovog programa koja nosi oznaku 5.0.

A Microsoft? Tu je priča malko komplikovanija. Naime Microsoft (dalje MS) je dobio počasno mjesto na ovom sajmu. S pravom i zasluženo. Windows 3.0 je bio produkt koji je naprosto dominirao sajmom. Nije bilo nijednog ozbiljnijeg proizvođača softvera koji nije imao verziju svog softvera za WIN 3.0 ili bar najavljivao ga. Poseban dio sajma bio je posvećen WIN 3.0 gdje su stručnjaci odgovarali na pitanja posjetilaca i usput prikazivali brdo aplikacija za dotične prozore. Sve u svemu, lijepo mjesto za biti, slušati i gledati, ali bilo je i ljepših mjesta. Mislim prije svega na one mašine kod kojih grafika nije gost već prirodan način rada.

## MERCEDESI PERSONALNOG RAČUNARSTVA

Jedan od njih je bio čuveni MEK. A tamo pod okriljem Apple kišobrana smjestila se sva gama Mak računara – i ne samo to. Na štandu Apple business centra iz Londona smjestila se australijska mašina kompatibilna sa Mekintošem! Stvar je utoliko čudnija ako se zna da je Apple sa svojom filozofijom zatvorenih mašina bjesomučno tjerao sve one koji su pokušali makar i malo da ponude zamjenu za Meka. Ne tako davno, dvojica momaka iz Njemačke ponudili su Aladin – dodatak za ST-a koji je emulirao Meka u potpunosti na ST-u. Stvar je ispočetka išla teško jer je trebalo nabaviti Mekove ROM-ove, ali su se ljudi snalazili. A onda je moćni Apple digao galamu i stvar je propala. Ovoga puta Apple pod svojim krovom nudi kompatibilca koji je u stvari laptop Mek! Stvar se zove OUTBOND i vaš je izvještač imao privilegiju da sa njom provede 2 sata oči u oči. Evo mojih impresija. Prije svega, računar je veoma malen i izuzetno dizajniran. Po dimenzijama je malo veći od A4 formata papira i debljina mu je svega 9 cm. Displej je prosječan za PC laptopove sa pozadinskim osvjetljenjem – rezolucija je 640 * 400 tačaka. Procesor je HCMOS verzija M68000 na 15 MHz. Tastatura se odvaja i povezana je IC zracima sa glavnim kućištem. Umjesto miša koristi se OUTBOND-ov pronalazak – ISOPOINT. Radi se o jednom dugom a tankom valjku koji se može okretati oko svoje ose i pomjerati lijevo – desno. Vjerujte mi da je u nedostatku prostora mnogo bolje od bilo kog miša. Za bateriju je izabrana standardna camcorder baterija koja traje 3 sata. Lijepo i standardno. Ugrađen je flopi koji čita i piše kako mekintoš 800K, tako i PC 1.44 MB format diskete. Te 40 MB hard disk. Težina je 2.5 kilograma sa baterijom i hard diskom. Sjajno. Jedina mana je to što se za romove morate snaći sami. Ali čim je firma nastupila pod kišobranom Apple-a, vjerovatno će i to biti uskoro riješeno.

A sljedeći je NeXT. Ovo je ubjedljivo bio najposjećeniji štand ovogodišnjeg šoua u Earl's Court-u. Potpuno crn – u obliku kocke (a zar je moglo ikako drugačije?) NeXT se predstavio sa novim verzijama kocke (Cube) i novim Next kompjuterom nazvanim NeXTStation ili od milja pizzabox (kutija za picu). Već na prvi pogled jasno vam je da u Americi prave daleko veće pice nego kod nas. A na drugi pogled, da NeXTStation nije ništa drugo do ista mašina u drugom pakovanju. Ali ne mari! To samo znači da kocka nastavlja da živi i da polako ali sigurno korača put uspjeha. Na NeXT štandu su osim samog NeXT computer Co. pojavili i Ashton Tate, Wordperfect i mnogi drugi. Od hardvera su predstavljene sve same poslastice. Zamislite grafičku karticu rezolucije 1120 * 832 u 16,7 miliona boja (megaboja) i sve to gonjeno sa i860. Naravno video ram je odgovarajući – pukih 4 MB. Sve te priče o digital signal procesorima i mrezama i ljubaznom operativnom sistemu, padaju u vodu kada sve to uživo vidite. Preporučujem prije gledanja NeXT kompjutera uživo da potražite čvrst oslonac za donju vilicu. To je naprosto nevjerovatno. Kada sa Komodorovog štanda na kome vrte spolove o multimediji dođete na NeXT štand na kome multimedija radi – razlika je više nego očita. A kad sam se jednom od demonstratora požalio da ja ipak radim samo DTP i da moram biti PC kompatibilan – pokazan mi je SOFTVERSKI emulator PC-AT-a na 12 MHz!!!! I taman kad sam polegao za novčanikom, pogledao sam u cjenik i vidio da još koju godinu neću biti kompatibilan sa ciframa pomenutim tamo. A evo i tih cifri, pa prosudite sami. Kocka sa printerom i monohromatskim monitorom (105MB hard i 8 MB RAM) košta 4385 GBP a ista kolor verzija 8255 GBP. Stvar je, dakle, slatka, ali...

## BILO PA PROŠLO

Šta reći na kraju? Sajam je bio i prošao. Lično ne vjerujem da ću otići na sljedeći jer je od evropskih sajmova opšteg usmjerenja CEBIT daleko najbolji. Osim toga, i Interbiro je ove godine bio veoma dobar (a što je još bitnije ima perspektivu). U svakom slučaju, ovo je bila još jedna prilika da vidimo gdje smo mi u svemu tome. I začudo utisak je mnogo, mnogo povoljniji nego prošlih godina. Gotovo sve viđeno ima se kupiti i u YU, cijene su dosta slične, a sa softverom jedino još malo šepa, ali biće i to bolje. Dakle ostaje nam samo da u predstojećoj godini svi zajedno polegnemo i da i mi damo svoj doprinos (koliko kod malen, ali ipak važan) YU računarstvu.

Autor zahvaljuje firmi ETD SyStems iz Sarajeva, Trg Pere Kosorića 1C, 071 – 650 – 633 i 650 – 462 na pomoći pri organizaciji ovog puta. Osim toga molio bih sve zainteresirane za cijene nekih modela i lične utiske da mi se obrate putem SEZAM-a – user name SSVITLICA.

Jovan Ćirilov

# OTISNUT NA ŠTAMPAČU HALSTON

Asistirala Zdenka Aćin

Hajte, molim vas, kompjuter kabalistički nazvan Abulafija u „Fukoovom klatnu" Umberta Eka jadan je i smešan kada se uporedi sa strašnim Korandorom, moćnim prirodnim računarom koji je zapravo glavni junak istoimenog romana Kristine Bruk-Rouz (Xorandor 1986)! Iz susreta dva mlada blizanca, strasna informatičara, s Korandorom proizlazi pravi košmar. Usled jedne greške u sintaksi kompjuter poludi i biće potrebna sva prirodna i tehnološka ingeniozna inventivnost blizanaca da bi moćni računar bio urazumljen. Roman je očaravajuća parodija kompjuterskog i nuklearnog trilera koji nam prikazuje jedan od mogućnih scenarija svetske katastrofe. To je parabola o modernoj komunikaciji, njenim obmanama i promašajima, jer Korandor nije ni više ni manje nego verna retranskripcija hiljadu i jedne jezičke procedure pod kojima se odvija komunikacija s njime. On je beskrajni niz integrisanih algoritama i struktura podataka u kojima se stokira i izvršava svaki pa i najbesmisleniji program. Jezik-mašina. Nebeski kompjuter. Ali spustimo se sa Jovanom Ćirilovim, piscem, kritičarem i upravnikom Jugoslovenskog dramskog pozorišta – na zemlju.

**Za pedante kompjuter je prava blagodet. Može rukopis da ispravlja do mile volje.**

*Svi već znaju da pišete pomoću računara. Kako to radite? Osećate li neku promenu u metabolizmu svoga pisanja?*

Savremeni francuski pisac Filip Solers u svom intervjuu beogradskom časopisu Delo izjavio je Jasni Šamić kako je za njega pisati kompjuterom isto što i „voditi ljubav s prezervativom".

**Sada svašta već znam da saznam pomoću kompjutera: tako važne i osnovne stvari kao što je, recimo, koliko je sati.**

Već u prvom trenutku učinilo mi se da je to vrlo duhovito, ali sam ubrzo shvatio da nisam razmislio šta je to Solers zapravo hteo da kaže. Možda da pisanje na kompjuteru znači da pisac takvim pisanjem ne doživljava zadovoljstva kad kad piše perom ili olovkom, baš kao što sa prezervativom ljubavnik u krevetu. Nije red da u razgovoru sa damom idem do kraja sa ovim poređenjem, čak i kada ono nije moje.

To bi isto tako moglo da znači da kompjuter, kao i kondom, sprečava oplođenje, te da je pisanje na kompjuteru osuđeno na jalovost.

*Ipak, kupili ste svoj „kondom"!?*

Kompjuter sam kupio kada mi je dojadilo da urednicima predajem neuredne rukopise. Koliko god puta prekucavao ispravljen rukopis, uvek se nađe nekakva sitnica koju bih želeo da ispravim. Za pedante kompjuter je prava blagodet. Može rukopis da ispravlja do mile volje, a da na kraju taj rukopis bude idealno čist. Ja ne kažem da sam baš klasični pedant, ali ima dana kada to jesam.

*I tada vam kompjuterske mogućnosti dobro dođu i pašu.*

**Ideja je da se sva pozorišta povežu, i da se na svakoj blagajni može kupiti ulaznica za bilo koju predstavu u mesecu.**

Kompjuterske mogućnosti najviše su me impresionirale kada sam prošlog proleća bio u Londonu. Tada me je moj dobri kolega Majkl Koveni, ugledni kritičar Financial Timesa, poveo na premijeru u Royal Court. Sa sobom je poneo svoj portabl kompjuter. Posle premijere pozdravio se sa mnom i otišao u gornje prostorije pozorišta, uništio predstavu i pritiskom na dugme poslao svoj napis direktno u štampariju. Dežurni urednik samo je pročitao članak na špiglu. Ja sam njegovu kritiku čitao u avionu sutradan rano ujutru. Razmišljao sam šta bih ja sve morao da uradim kada bih posle predstave morao da za jutrašnje izdanje Politike predam rukopis. Prvo bih morao da jurim kući da na tadašnjoj bednoj pisaćoj mašini, sa lošom trakom, napišem kritiku. Tada bih uzeo taksi i nosio napis dežurnom noćnom uredniku. Pri tom strepeo da li je noćni portir jedan od onih koji me zna kao dugogodišnjeg saradnika lista ili ne. Ostala bi mi još jedna strepnja: da li je slagač tačno složio moj rukopis i da li je korektor ispravio njegove greške.

Mislim da sam te večeri odlučio da kupim kompjuter, i to lap top ili portabl, da bih jednog dana i ja iz neke tuđe kancelarije mogao da srubljujem njihove predstave. Uveren sam da će nastupiti uskoro dan kada će listovi i časopisi za koje redovno pišem, NIN i Politika, moći da primaju moje članke preko mog broja telefona. Za sada sam, i pored moje kompjuterske revolucije, svoj sopstveni kurir.

**Moja zaraza kompjuterom dovešće do kompjuterske revolucije na mom radnom mestu u Jugoslovenskom dramskom.**

*Na koju ste se mašinu polakomili?*

Kupio sam SHARP PC 4500. Nisam imao para da odmah kupim i štampač. Svoje prve brižljivo napisane reči na kompjuteru, a osećao sam se kao Gutenberg, kao da sam lično izmislio računar, nosio sam zajedno sa svojim laptopom kod prijatelja i mučio ih da mi štampaju moje prve napise u novoj tehnici. Mučio sam se ja, mučili su se i oni. Morali smo da najpre isključujemo njihov kompjuter, pa da uključujemo moj, da trošim tuđu hartiju i dragoceno vreme. Na nesreću neke moje loše komande učinile su da jedna strana bude ispunjena tekstom, a na drugoj se kočoperio usamljen broj stranice. Ostali deo teksta bio je prazan. Moj prijatelj bio je na granici da psuje, ako ne baš mene ono moj komjuter, no ipak se uzdržao.

*I jeste li se rešili muke?*

Pre neki dan kupio sam štampač EPSON LX 400. Bio sam još nešto dužan za svoj hardver, kada je u jednu privatnu radnju stigao štampač baš takav kakav mi treba. Mom poveriocu sam obećao da neću kupiti štampač dok se kod njega ne razdužim, ali prodavac kompjutera, mladi inženjer, mi reče da ima ograničeni broj štampača i da ko zna da li će uslovi uvoza ostati isti. Te je u subotu u pet do dva odoh u banku, skidoh sa računa poslednje pare, učinih greh protiv savesti, ali imam štampač. U radnji mi je vlasnik pokazao deset minuta kako se uključuje štampač, gde se šta štrati, gde spaja, koje dugme pritiska i ja sam sav znojav i zabrinut došao kući i odmah počeo da uključujem štampač dok ne zaboravim što mi je na brzaka bilo pokazano. Sutradan sam slučajno sreo mog poverioca koji mi je ironično nasmejan rekao: „Čujem da ste kupili divan EPSON." Opet sam se preznojao. Nisam znao da se kompjuteristi svi međusobno znaju i da pred njima nema tajni. Rekoh da mi pritiska ne savest, a on odgovori: „Pa savest i postoji da vas pritiska!"

**Posle premijere otišao je u gornje prostorije pozorišta, uništio predstavu i pritiskom na dugme poslao svoj napis direktno u štampariju.**

*Pametni kakvi ste mače, bez sumnje ste brzo savladali početničke lekcije!?*

Moj prvi susret sa kompjuterom bio je pravi košmar. Posle jednog sata instrukcija otišao sam kući i pokušao sam da nešto napišem. Najpre nisam umeo da pređem na ćirilicu. Sve bi bilo u redu, ali je umesto č, ć, š i ž ta glupa mašina ispisivala recimo ‹ … ili bih sve otkucao kako sam zapisao komande, a svaki put bi mi izlazilo na ekranu bad command, try again! Pokušao bih po drugi put, isto. Da čovek poludi. Shvatio sam: ja glup, a on tvrdoglav, i nema tu spasa. Pokušavao bih bezbroj puta, isto. Ostajao bih do kasno u noć (a ja ležem relativno rano, u ponoć), negde do tri sata ujutru, i nikakvog rezultata: bad commmand! Jedva sam čekao da svane i da telefoniram svom instruktoru i pitam da li sam ja to glup ili mi se možda već pokvario moj novi kompjuter. Obično bi se ispostavilo da je greška što nisam udario neku tačku ili rastojanje. Sitnica, ali za glupu, gvozdenu spravu dovoljno da mene napravi glupim. Bilo je trenutaka kada sam hteo da uzmem sekiru i na paramparčad isečem tu đavolju spravu. Srećom, sekira mi je već godinama zaturena negde u podrumu.

**Uveren sam da će nastupiti uskoro dan kada će listovi i časopisi za koje redovno pišem, NIN i Politika, moći da primaju moje članke preko telefona.**

*Ali, glupi uče iz knjiga!*



Snimio: Milenko Stefanović

Kada sam se malo uputio u rad na kompjuteru došla je laza kada sam za stvari koje ne znam oko kompjutera kad god bih u društvu čuo da neko ima kompjuter ili ne daj bože (po njega) da je kompjuterist počeo da postavljam pitanja bezbroj pitanja koja su zaista osnovna, osnovačka i za nekog velikog stručnjaka smešna To je trajalo dok nisam saznao da u „Tehničkoj knjizi" u njihovom izdanju postoji knjiga „Chi Writer" koju većina Jugoslovena pogrešno izgovara či-rajter umesto kaj-rajter Autor priručnika lepo je objasnio mnoge elemente rada na kompjuteru za koje sam morao do tada da zapitkujem brojne žrtve u slučajnim susretima

Ne razumem baš sve ali mnogo štošta uspem da sebi protumačim sada iz knjige

**Za sada sam, i pored moje kompjuterske revolucije, svoj sopstveni kurir.**

*Nešto vam valjda kaže i sam program za pisanje?*

Sada svašta već znam da saznam pomoću kompjutera tako važne i osnovne stvari kao što je recimo koliko je sati koji je datum i razne karakteristike mog računara Ali najviše volim da na ekranu izazovem sliku klavijature i to svih tipova slova To izgleda kao neki raskošni persijski tepih Kada nekom hoću da demonstriram rad mog sharpa – ja mu to prikažem Katkada se obrukam ali u većini slučajeva klavijatura se pojavi u svoj svojoj raskoši

**Pokušao bih po drugi put, isto Da čovek poludi Shvatio sam: ja glup, a on tvrdoglav, I nema tu spasa.**

*Ta raskoš može biti i kompjuterska bolest Čuli ste za „virusnu" zarazu?*

Moja zaraza kompjuterom dovešće očigledno do kompjuterske revolucije na mom radnom mestu u Jugoslovenskom dramskom Ovih dana sam od nekih vanrednih para poručio kompjuterski uređaj za ulaznice na biletarnici Umesto šoka koje pozorište doživi kada se formira mesečni repertoar tamo negde krajem meseca za naredni mesec i kada valja što pre iskucati bezbroj ulaznica po blokovima koji su obično na izmaku i jako skupi za ponovno štampanje sada će ceo sistem biti u računaru Biletari će imati na ekranu koje je ulaznice prodao a koje nije i divota! Ideja je da se sva pozorišta povežu i da se na svakoj blagajni može kupiti ulaznica za bilo koju predstavu u mesecu Naručio sam i terminal za moj biro tako da ću u svako doba dana moći da saznam koliko je prodato ulaznica za predstavu te večeri a ne da moram da to saznajem specijalno podešenim ljubaznim glasom od biletara koji obično krije pravo stanje stvari ako je sala praznjikava

*Hoće li i same predstave biti ubuduće kompjuterizovane?*

Kompjuterizovaće se i računovodstvo JDP Za pozorište je važna i statistika a ne samo ulaz i izlaz To da nemamo nikada para to znamo i bez kompjutera ali za pravljenje mesečnog repertoara neobično je važno znati kretanje publike za svaku pojedinu predstavu Da li posela za jednu predstavu raste ili opada? Da li bolje ide domaći komad ili klasika? Od toga zaista treba da zavisi šta će se narednog meseca staviti na repertoar Pomoću kompjutera kažu može da se brzo saznaje mogućnost paralelnih podela novih komada u odnosu na postojeći repertoar Pozorište koje kao JDP ima tri scene mora da zna šta se sa čim može uporedo davati Naravno da je to moguće raditi i bez kompjutera i radi se i sada onako peške ali drugo je kada se to brzo sazna pomoću mašine A i mogućnost grešaka je manja

*Neki greške ne vide Greške ih hipnotišu*

Moja majka neobično voli kompjuter Nisam uveren da će u svojoj osamdesetoj godini učiti da na njemu radi mada me ni to ne bi čudilo tako je moderna Ona voli kompjuter iz drugih razloga – primetila je da me on vezuje za kuću a željna je da budem kraj nje Istina ne pričam bogzna koliko s njom dok radim za ekranom pa ipak tu sam kraj kompjutera kao nekada seljaci kraj toplog ognjišta

*I Šekspir je slično mislio Govoreći o zamamnom plavičastom zračenju predvideo je slučaj nalik vašem Odviše se izlažete moćima zračenja*

Pre neki dan sam čuo da ne treba dugo raditi s kompjuterom jer se čovek izlaže zračenju Neko je o tom govorio na televiziji Opet sam u društvu od čoveka koji se u to razume saznao da je zračenje od kompjutera tokom celog života manje od zračenja kosmičkih zraka prilikom jedne jedine vožnje prekookeanskim avionom Ja sam prosto rešio da o tom ne vodim računa Ljudski život se, ipak, toliko produžio zbog higijene i drugih pogodnosti savremenog života da nešto mora da ti i oduzima i troši taj život

*Da biste to na drugoj strani nadoknađivali*

Isto tako kažu da treba s vremena na vreme zbog očiju gledati u svetlo zelenu površinu papira ili zida To bi moglo biti istina ali još nisam isprobao Veći mi je problem što moram da zamračujem prostoriju u kojoj radim da bi mi ekran bio čitljiv

**To da nemamo nikada para, to znamo i bez kompjutera, ali za pravljenje mesečnog repertoara, neobično je važno znati kretanje publike za svaku pojedinu predstavu.**

*Mislite na reči? Što biste ih čitali? Kad već imate „računaljku" pišite naslepo*

Da se vratim na početno pitanje – kako doživljam samo pisanje na kompjuteru u odnosu na druge tehnike Tu nesumnjivo ima razlike u načinu mišljenja Siguran sam da pesme ne bih mogao da pišem na kompjuteru kao što ne mogu ni da ih pišem ni na pisaćoj mašini Pisanje eseja kritika i svakakve druge proze sasvim lepo ide na kompjuteru Kako odmah ispravljam greške jer je to tehnički moguće, pisanje ide sporije Još ne znam da li to utiče na stil i način mišljenja prilikom pisanja Možda je misao pročišćenija ali za sada može da se desi da ta opreznost može neku misao da potisne sakrije i zagubi Ali sa većom uvežbanošću siguran sam da ću doći do nivoa da misao može da se neposrednije izliva u tekst kao da savršenog tehničkog posrednika i nema Ali ko zna možda će ostati neka nijansa koje nikada neću biti svestan Gotovo je sigurno da misao ne ide isto sa guščijim perom i elektronikom Ali što naša misao kompjuterske ere ne bi ličila na spravu po kojoj je dobila ime?

*Menja se i era u sebi, pa i sprava biva sve složenija i zavodljivija*

Plašim se da ću, taman kada se naviknem na Sharp PC 4500 i na Epson LX 400 poželeti da idem dalje u nešto skuplje i složenije Ni sam još ni naučio kako mi se ta sprava zove Često pogrešim i na pitanje kakav štampač imam – kažem „Halston" Na šta mi poneko s pravom uzvrati „Pa to vam je muški parfem, a ne printer!"

### Post scriptum

*A meni su kada sam se hvalio svojim Epsonom LQ 550 nudili PostScript budzašto za cigio 10 hiljada DEM! Svašta poštanska skriptgerla! Šta će mi rekoh kada postskriptume ionako nikada ne pišem Po tome smo srodni računari su za nas pun tajni*

Razmišljam kolike mi samo tajne još krije Kada pritisnem izvesne komande još mi iskoče neke labele koje ne umem te čitam ili mi izbriše pokoji red Sada se samo plašim da zaboravim da memoriram ono što sam napisao Ode mast u propast!

# EDGAR,KOMPA,MI

et toutes les autres
**gegen den Rest der Welt**

Novica Milić

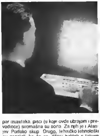
### Help?

Pred (ili sa) kompjuterom imam dve fantazme, skoro halucinacije. Prva: moje „ja" je bezuslovni deo njega. Ja sam samo jedna tačka, vrlo mala, sićušna, skoro i izgubljena negde u pedesetom ćošku stotog pod-direktorijuma tvrdog diska. Ta se tačkica (jedva bajt, možda pre bit ili dva) kad god uključim mašinu, nasumično učitava negde između bafera konzole (ekran, tastature) i ostatka RAMa kao „floating point" (kako to prevesti?); imam osećaj uspinjanja, čak uspenja, i samo sam delić koji opslužuje veliki Svemirski Računarski Centar. Druga, prvoj suprotna (i logikom fantazme koherentna) znani je arhetip: svi smo, pred računarom ili bez njega, samo uobražene mrdalice (softveri koji su sebe prozvali ljudima) unutar Marko Haosmosa u kojem nas, bez prestanka, asemblira i disasemblira neki ludi haker, kod kompjuterskih laika poznat pod imenom „Bog".

Da li se nakon tolikih decenija, i naročito nakon poslednje, kad su ove mašine postale i „lične" („personalne"), pitanje promenilo i postalo, iz onoga *Kako se služiti kompjuterom?* u — *Kako se kompjutera lišiti?* Da je računar postao ili postaje isto što i usisivač, metla, dubravnik, igla i lokomotiva u isti mah — ukratko točak za treći milenijum — u to je sve manje sumnje. Da je koristan, to je opšte mesto. Ali da li ćemo svi završiti buljeći povesdan u monitor — verujući i prevarajući se u opslužioce soft-hard-vera unutar Mikro ili Makro Hakerske Manipulacije — to je već kao problem stiglo na dnevni red. Pošto se ne možemo lišiti kompjutera („da li se lišiti" je sasvim nesuvislo), *Kako?* postaje neobično važno. Gde možemo bez kompjutera? Kada? Kad na to odgovorimo, znaćemo i da ga koristimo u nekorisne svrhe, za igru koja će nam pružiti odgovor na glavno pitanje, na pitanje svih pitanja: Šta su to — kompjuteri?

### Padajući ili podižući meniji?

Pokušaću da vam — razume se, koliko to moje skromno iskustvo dopušta i samo u pojedinim elementima, nikako svim — opišem kompjuter sa stanovišta književnika. Prvo i osnovno: priča, pesma, drama, esej ili prevod mogu se pisati i prstom po pesku. Ako ste kojim slučajem naumili da dobijete Nobela, postići ćete to i mastiljavom olovkom. Ali pisci ne bi bili to što su (noćna i dnevna gundala, gutači rečnika i čitači nevidljivih međuznakova) da ni su večito zaintrigirani i alatkom kojom proizvode svoje uzvišene žvrlje. Velim — zaintrigirani. Kad se među tom rajom pronela Dobra Vest (da ove mašine mogu i njima biti od vajde), u njih se uselila dvostruko vezana spletka. S jedne strane, to treba probati. S druge strane, rezervisani kao i uvek naspram tehnologije i čitave njene ideologije progresa (za pisce kao pisce progres je reč bez posebnog smisla, u književnosti se ne „napreduje") oni na kompjuter gledaju s podozrenjem i distancom većom nego kod drugih, stvarnih ili mogućih korisnika jedne naprave koja zuji i nervira. Prvo, osim



par izuzetaka, pisci (u koje ovde ubrajam i prevodioce) siromašna su sorta. Za njih je i Alanjev Portfolio skup. Drugo, tehničko-tehnološki su zaostali; ko će se pitaju baktati s takvim „pojmovima" kao što su *LPT1*, *COM2* ili sačuvaj nas Bože „samostojeći oduševljivač" (kako je jedan od njih preveo *autodesk animator*). Treće, računarski su polupismeni; kad im odnekod padne na sto PiSi, Mekintoš ili, možda „za-u-vrh-krila" (*lap-top*) Tošibice, onde se dugo i pobožno čude, i komšijski klinac, u čije vesele prste zazorno poveravaju svoje buduće ljubimče, izgleda im kao i ti vanzemaljac. On im uostalom donosi i prvi ČiRajter, i ostavlja ih, uz zbrkane naredbe koje im je nažvrljao na poleđini nekog nemačkog priručnika (sa zastrašujućim naslovom „*Einführung in MS DOS und Rechnersbenutzerhandbuch*" uh!), da dugo u noć uvežbavaju ulazak i izlazak iz onoga što će, mnogo docnije, znati učeno da zovu „Kaj Rajter".

O da. Naši pisci imaju i problem koji njihove zapadne kolege nemaju. Taj problem se zove naša slova (čćžš pa i đ). ASCII i ANSI imperijalisti nisu mislili na ovu jugo-zabit. Programi — editori i procesori teksta — uglavnom su na engleskom. Pisci uglavnom znaju engleski, ali softverski engleski je poseban slučaj. „Džastifaj? Šta je to? A... Ne, ne treba mi to. Serč and riplejs? E, to to!" Pošto im dosade krive kuke mesto naših slova, a ni TeTri (učeno „TeKjub") ih nije bacio na nos, i pošto im je majstor Vlada — oh, s kakvim strahom svoju grafičku karticu nose u servis! jer šta ako majstor Vlada zabrlja pa im uništi Kompicu? — rešio večiti problem domaće abecede (azbuku da ne pominjem), onda već probaju takva nebeska čudesa kakva su razni, i preko komšijskog klinca (stalno se pitaju zašto ih u poslednje vreme zaobilazi preko drugog ulaza?) dobavljeni Vordovi. „VordStar je super!" „Ma ajde. Nisi video VordPerfekt!" „To je za sekretarice. Stvar se zove prosto Vord, Majkrosoftov, pa za Vindouz."

Prolaze nedelje, a nedelje čine mesece. Ponosni vlasnik nekog „Nejm" (to je negdašnji „NouNejm") kompatibilca napokon na svom tvrdom disku ima nekoliko verzija nekoliko Vordova, uz Venturu, fontove i font-editore to je ono čime mu je ispunjen računarski život. Doduše, kao pisac, ima problema što novele započeta u preistorijsko vreme pisaće mašine nije otad odmakla ni za pola rečenice. On ju je ukucao u mašinu. Ali kad treba da nastavi, i kad izjutra ili izvečeri uključi aparat, put od „rut" direktorijuma do kataloga /MOJVORD/MOJTEKST/MOJANOVELA mu je odviše prepleten opasnostima koje mogu imati različito ime, ali se uglavnom svode na „nov_od_Mikija_presnimljen_program_čeka_da_se_vidi" ili, za maštovitije, na osamnaestom nivou sinoć ostavljeni Sokoban. (Ko bi rekao da će najinteligentnija igrica biti guranje sanduka po lavirintima ekrana!) MOJANOVELA čeka, na neki budući, zgodniji čas, kad će Pisac već shvatiti da osim padajućih i podižućih menija pod kapom Herkulesa postoje i takve zgodne alatke kao što je „Najmanji Najbolji Program za Pisanje", zvani „Norton Editor", i da je on, za sve praktične svrhe (a i šire, ili „respektivno", kako bi rekli programeri), sasvim dovoljan. Tada dobijamo ono što se može prozvati Pisac S Kompjuterom, a da taj naziv nije šuplji niz tri inkompatibilna, poluraštrojena, nakostrešena i uglavnom neuposlena čudovišta.

### Insert/Owerwrite?

Piscu Na Kompjuteru — toj novoj i istorijski čudno nastaloj sojti — najveći problem je psihički. Pisati omiljenim perom, kupljenim u Parizu podno Monparanasa za stotinak franaka, pisati po papiru, belom ili osenčenom nekom od boja koje je i Prust koristio — i pisati „u televizor", tj. kucati i buljiti u fosforaste piksele nekoliko sati na dan ili na noć, nije, razume se, isto. Onima koji su davno prešli s ruke na mašinu (pisaću), to je možebit lakše. U osnovi, izgleda da je problem i više nego psihološki (i pre za psihoanalizu jednog Lakana, Deleza ili Nikolasa Abrahama): ruka je prva alatka, a proširiti tu alatku na tastaturu vezano je za adaptaciju na tzv. fantomske organe. Ljudi koji pišu nikad, po pravilu, nemaju doživljaj vlastite ruke kao posebnog organa. Pisac je zaista Telo-bez-organa, prostor ili površina koja ostavlja utisak da se piše „iz sebe", „iz stomaka". Stoga je prelazak na kompjutersko pisanje bolan, dugotrajan, ponekad iscrpljujuć, mnogi odustanu i pre iskušenja. U tom je prilično krupna razlika između upotrebe kompjutera u slučaju pisaca i svih, ili većine, drugih.

Ta razlika nije jedina. Pisci ne kupuju kompjutere da bi vodili „mejl-merdž" prepisku, gomilali „di-bejz" datoteke, „kam-kedili", „bejzikovali" ili činili slične i hvale vredne računarske podobštine. Njima kompjuter treba „prosto za pisanje". Kako ta vrsta mašina nikad ništa ne radi „prosto" (makar se tako činilo, i što složeniji njen posao bio, to je jači utisak jednostavnosti), a kako je od asamblerskog listinga jedi-



no komplikovaniji pesnički stih ili prozni zaplet ovo „prosto za pisanje" nosi u sebi, za ljude kojima su tekstovi znakovi reči i sve ono između reči sudbina, jedan skoro bezimen problem. Nazvaću ga filozofskim, uz dabl andersajn svih rezervi koje se u našim geografijama i istorijama mogu imati prema apstraktnom i prividno beskonačnom umovanju. I pokušaću da taj problem najkraće što mogu opišem u narednom odeljku.

#### Kalkulus i njegov elektronički muskulus

Šta će biti s kompjuterima, a verovatno i sa nama uz njih, najmanje će zavisiti od toga da li će se proizvesti biočipovi, koja će biti interna arhitektura procesora i da li će imati i kakvu softversku podršku. To su uglavnom tehnička pitanja (a zapravo većim delom stvar odluke krupnih biznismena). Šta će biti s kompjuterima, i sa nama uz njih, zavisi pre svega od toga da li ćemo i kada razumeti šta su to — tzv. kompjuteri. Na to pitanje ne daje odgovor tehnologija, već filozofija tehnike (i filozofija uopšte). Odnosno, da budem precizniji: filozofija procesora.

To — ono što sam upravo krstio kao „filozofiju procesora" — bilo bi filozofsko promišljanje prošlih, današnjih i budućih mašina za obradu informacija. Rekao bih rado da je to „filozofija kompjutera" (tout court), da mi barem dve sumnje ne zadržavaju taster. Prva: kad sam pre otprilike godinu dana hteo da priredim blok tekstova o „književnosti i kompjuterima" za jedan časopis (slično ovome što sada ostvaru je koleginica Aćin), glavni i odgovorni ovih *Računara*, gospodin Jova Regasek, na pitanje o filozofskim doprinosima temi zavrteo je, a i ja za njim, setno glavom. Nije da takvih rasprava nema, ali su još na periferiji pantehološkog razmišljanja o računarima. Druga sumnja: mi (pod ovim „mi" mislim na sve one korisnike ili iskorišćene kad su soft/hardveri u pitanju) još nismo načisto koje je najbolje ime za mašinu za kojom sedimo. Za tehničare i inženjere to je, razume se po sebi, „kompjuter" u prevodu „računar". Jer ta mašina, Bože moj, ne radi drugo do li — računa. Ako nekom tehnoinženjeru kažete umesto reči „bajt" reč „oktet", velika je šansa da vas neće razumeti. Kompjuter kao bajagi govori engleski, ali, onda kad progovori, na primer, francuski, onda on za „bajt" kaže „oktet", kao što na francuskom, on sebe po pravilu ne zove „*computeur*" nego — „*ordinateur*". I to ime „*ordinateur*" — upućuje daleko više na smisao sređivanja i uređivanja, na poredak i preuređivanje, na „aranžiranje", nego na kompjutaciju u uobičajenom smislu rada s brojevima (računanje, kalkulacije, kalkulusa). Verovatno bi najbolje ime bilo — procesor, odnosno, dugim i pomalo dosadnim nazivom — aparat (mašina, stroj) za informatičku obradu. Ne, dakle, „za automatsku obradu podataka" (relativnost ovih „automata" je notorna), pa ni „za obradu podataka" (jer je objektni genetiv ovde zavodljiv), već za „informatičku obradu", gde ne obrađujemo samo informacije, već gde i druge informacije obrađuju informacije, sebe pa i nas.

Svakako, nije najvažnije kako ćemo ih zvati, već kako ćemo ih opisati da bismo bili svesni sa čim imamo posla. (Moj sam nedavno nazvao po jednom kompjuteru koji se u filmu napio šampanjca i pobrljavio, da bi se na kraju sklonio pred gazdom poslavši na sebe preko Tokija stotinak hiljada volti u času dok mirno čeka ovaj elektronički sepuku, on jedino zamera svom vlasniku: „Nikad me nisi ni upitao kako mi je ime!") Jer, ako je procesorska mašina za informatičku obradu nešto više ili nešto manje — a ipak nešto drukčije i ne samo — „računar", onda valja drugačije shvatiti ono što se zove „kalkulacija" (račun, matematika koja je pre algebre i geometrije bila *mathesis*, učenje) i posebno ono što je ključno ime za „*science of computing*", naime — algoritam. Ako čitate Knutovu enciklopediju — *The Art of Computer Programming* — shvatićete da dve operacije koje leže u temelju dosadašnjih, a sva je prilika i budućih „kompjutera", nisu „saberi" i „oduzmi" (odnosno njihove varijacije), već — premesti i uporedi, tj. premesti u skladu s izvesnim otporom. Misao o premeštanju i otporu nije, uz sve poštovanje tehnoinženjera, tehnička niti tehnološka misao, niti je pak matematička misao u užem i strogom smislu te discipline. Treba li da vam kažem da se oblast koja o informaciji misli u kategorijama otpora i premeštanja zove — retorika, a da misao koja pre svega u tim kategorijama razmišlja o ljudskom subjektu zove — psihoanaliza?

Verovatno da treba. Znam da sugestije — poput ovih „retorika" ili „psihoanaliza" kompjutera — zvuče isuviše šokantno da ne bi izazvale skoro nesavladiv otpor i bile obično na samom pragu poruke odbačene („Čuj! On oće da psihoanalizira Kompu!" „Ma još gore! On bi da ga shvata kao retoričku figuru!"), ali one nam mogu svojim udarnim (šokantnim) nabojem, pomoći da problem ovih mašina premestimo na teren u kojem su neki odgovori izgledniji.

#### Abort, Retry, Ignore?

Poslužimo se tim premeštanjem jednim od onih mnogih kojim nas uče asemblerske instrukcije i pseudoinstrukcije. Bez namere da sve kažem (i bez mogućnosti da to sada na ovom ograničenom prostoru i pokušam), evo nekoliko naznaka, u obliku pitanja.

*JMP FAR*. Prvi skok je krupan, obično najvažniji i skoro presudan. To nije pitanje „šta su kompjuteri" (šta su po sebi, kao silicijumske spone, elektrike novog doba etc.), već šta su — za *nas*? Ko smo mi kao subjekti (bića obdarena, kako kažu, čulima i mozgom) koji se nalaze u nekom momentu svoje istorije pred procesorima? Da je njihova pojava donela novа okruženja (nov rečnik, nov raspored vremena, nov način utrošnje ljudske energije, nov način mišljenja, čak — novu modu), to nije sporno. Koliko nas oni menjaju duhovno? Jer, ukoliko mogu (dobro, ovde u Jugi to je čista teorijska mogućnost) da pritiskom na nekoliko dirki skočim za tren ili dva preko Velike Bare u Kongresnu biblioteku, to onda bitno menja moje poimanje informacije, ali i prostora-vremena, tzv. hronotopa koji je jedna od ključnih shema svačijeg života.

*JMP NEAR*. Drugi, prividno bliži skok — s novim hronotopom, šta se zbiva s tom singularnom tačkom koju svako od nas nalazi manje više u središtu sebe i sveta i zove se „moje Ja"? Je li u novom prostoru vremenu, u novom rasporedu mesta i trenutaka to „Ja" (moje, tvoje, a kolektivno naše, vaše, njihovo itd.) na istom mestu, u istom središtu? Jesmo li, kako bi se to po naški reklo, još uvek „pri sebi", „kod kuće", u svome „Ja"? Što ne znači da smo ikad bili, već da se bliže ili dalje pomeranja, često jedva uočljiva, koja se zbivaju sa ljudskim subjektom, uvek odvijaju u zavisnosti i preko tog „Sebe", onog „Ja", čije mesto nije zauvek fiksirano i čija korenita premeštanja mogu dovesti i od — izmeštanja do gubitka ili barem zalamljenja perspektive, a time i posmatranja sveta. Preterujem? Možda. Ali zamislite jednog Josifa Visarionovića priključenog na server Čeke i njegovu svesnu ili podsvesnu grešku — *lapsus taslaturae* — gde umesto *Alt-X (Quit Program)* pritiska *Ctrl-X (Destroy Every Enemy For Ever)* — Oduvek sam bio za jednostavne programe, barem kad su *Ctrl* i *alt* u pitanju.

*JMP SHORT*. Još bliže, ne samo da novi načini daljinskih upravljanja, arhivizacija, kontrole trećih mašina preko drugih i prvih, otvaraju pitanja koja su bila sasvim nepoznata, dosadašnje navike i odgovore — odbrambene ili napadne reakcije — čine katkad izlišnim, a često pogrešnim. Tehnologija (oduvek) iznova raspoređuje prikupljanja i trošenja (resursa) energije, ne samo fizičke i ne samo intelektualne, već i one koja je prividno u emocijama, a efektivno u podsvesnom. Tehnologija nema ima funkciju odvraćanja (jer predstavlja i proizvodi otpor), kao god i funkciju preusmeravanja: nemam vremena i moram uprošćavati, pa stoga navodim samo primer velike Industrije Odvraćanja i Straha (vojne, tj.), u kojoj se rađaju monstrumi popot kompjutera. Funkcija preusmeravanja računa između ostalog na produkciju pozitivnih emocija, na prizivanje *radosti što se može uspeti*. Ništa nije pogubnije od tzv. uspeha. Pojačano osećanje da ćemo sedeći za monitorom „uspeti" više nego što smo „uspeli" na druge načine dodaje lancu naših neuspeha još jednu karіku: da nećemo nikako „uspeti" ukoliko *baš tako* ne budemo „uspeli". U tom skučenom solizmu, u obrtu koji ništa do retoričko (retorika mišljenja je skriveni-



ja moćnija i samim tim daleko presudnija nego tzv. obična retorika, besedništvo, govorancija), kruje se onaj kratak spoj što izvestan kolektiv dovodi u stanje permanentne neuroze do šizo-stanja uspeha ili neuspeha. *Ili* — *ili*, nama treseg. *Ili* ćemo uspeti u napredovanju, *ili* nas neće biti. Katastrofa koja vreba iza ovog *ili-ili* leži u tome što (moram ponovo podvući, za šta se izvinjavam) Uspeh shvaćen po shemi Napretka ne zna za druga uzmicanja osim taktičkih. U ime Uspeha Napretka sve će se učiniti, pa i ubiti (teško je ubiti u ime Boljeg, ali lako, vrlo lako u ime Naprednijeg). — Drugim rečima, i da završim ovu tiradu u malo bajtova, enormni potencijali otvoreni pronalaskom procesorskih mašina imaju tu zvrčku što slabe naš emocionalni i podsvesni imunitet, i otvaraju socijalna i etičke brane takvom brzinom da se opasnost ne može dovoljno hitno preduprediti novim, i novom hronotopu (čitaj: okruženju) primerenim sredstvima. Kompjuteri donose novu brzinu donošenja odluka, takvu da mi nismo više dovoljno brzi za njih, otuda sasvim realne opasnosti da im se prepusti da donose odluku o sebi samima i o nama. Panično? To samo znači da ne čitate P. Vinlija, Ž. Bodrijara, ili, pak, analitičare Pentagona. Ili da niste bili u Njujorškom Sitiju, među onih desetak oblakodera krcatih računarima, u kojima je upisano 95% svetskih finansija, i u kojima, na primer, ekonomija jedne zemlje može, ukoliko se neki soft-hardver napije šampanjca, *„in zero wait state"* biti svedena na golu nulu? Ili zar zaista mislite da je vaš Edgar jedini koji svojim pametnim okicama motri na svaki vaš pokret? Da su strategijska igra samo igre? Zar niste čitali barem Orvela?

## Do not panic! Press F1 to get help.

Taj se help dobija na drukčijem dugmetu. Pisci, da se posle ovog izleta opet vratimo njima, ako i ne znaju tehnologiju sijalice, znaju za jednu drukčiju tehnologiju čitanja i prebiranja po znakovima vremena koje je samo jednom nogom to, a drugu izbacuje ispred, u tzv. budućnost. Ukucavajući gornje redove pada mi na pamet da jedino pisci (sada uključujući i one koji iz ko zna kojeg razloga to nisu postali, ostajući dobri čitaoci) imaju sluha za nadtehnološko razmišljanje o kompanije. Bio bih nesrećan ukoliko ijedan čitalac *Računara* shvati da išta imam protiv računara: imam protiv izvesnih upotreba računara, a to je sasvim drugo. I zato mislim da je pitanje *kako se lišiti kompjutera* u nekim oblastima prvorazredno. Čitamo mnogo o novim tipovima monitora, o novim sklopovima čipova, o novim dosetkama programera, i to je dobro. Ali ne čitamo (jer o tome i ne pišemo) o novim upotrebama svega toga u državi nam, ne raspravljamo o tome ko, kad, gde i kako ima pravo ili dužnost, ovlašćenje i slobodu da o nama unosi, obrađuje ili koristi podatke. Znamo vrlo dobro da je jedno usamljeno kompjutersko ostrvce — što je 99% nas — kap koja sama nad sobom isparava, i, uprkos tom saznanju, ne vodimo žučna rasprave o tome kakav bi bio, recimo, program jedne Stranke Za Računare (ili, barem, Lobija kompjutensta), koja bi nas izvela (a ima nas oho-ho!) pred zdanja neke od naših brojnih skupština da urlamo protiv varvarskih carina koje plaćamo, protiv domaće šmalciger industrije i njenih povlašćenih astronomskih cena (kao da smo im najveći tuđinci!), i za informatičko bratstvo-jedinstvo (što će reći za princip jedan čovek — jedan modem) umesto za ekskluzivistički jupak-pupak sistem čiji je jedini mogući izgovor očajna ili pokidana PTT mreža. *Chaos Computer Club in Belgrade?* A zakaj ne? Čitamo, najzad, u stranim kao i domaćim Bajtovima o svetom kopirajt pravu velikih softverdžinica, ali ne i o još važnijem pravu klase lumpenproletarizovanih individualnih osvisnika na izvorni (*soruce*) kod. Kad se adaptira Šekspir za pozorište, onda niko ne poleže autorska prava, ali softverska moltinacionala je jedina malija koja je uspela da vam po skupe pare proda nešto za šta prethodno samo slutite šta je, nešto što potom ne smete ni za tačku da promenite (!) i na čemu, najzad, redovno stoji upozorenje „Svaka greška je samo vaša! Mi ne snosimo nikakvu odgovornost". Ne znam branše koja se tako brzo izdigla među bezgrešne anđele kao što je to softverska. I ne znam za gnusniju radost od one koju su sasvim nedavno iskazali neki domaći kompjuterski mislioci što se kod nas otvaraju agentura belosvetskih softverdžinica koje će nam navodno prodavati svoje mačke u domaćim džakovima, a zapravo denuncirati pirate. Zar niko ne shvata da je nekoliko naših pirata za razvoj jugoinformatike učinilo više nego zakonom pokriveni „nosioci razvoja"?

Dokle god nakaza „pravne" države ima geslo — pirate u zatvore da potom provali domaća softverska ingenioznost na svakom poljančetu (stara pesma: siromašni su pametniji) — dotle sam za anarhiju Fejerabendovog tipa. I dokle god nas tlače, i dokle god je tehnologija samo opijum za mase, dotle se treba vladati kao što su se trudili pisci: imati svoju „francusku sedam" i trijati iz dana u dan, na smenu, ili još bolje — rezidentno — protiv režima. Naročito ovog.

## Alt-F8 (u Norton Komanderu): History

Ne čitamo još o jednoj važnoj stvari: o istoriji kompjutera. Temi „književnost i računari" pripada i osvrt na priče o tome kako su računari nastali, kako su se menjali, odakle dolaze i kuda idu. Znanja o tome su najčešće površna, a među ekspertima vlada pometnja. Čitao sam nekoliko knjiga iz istorije ovih mašina. Među onima koje ne prepričavaju poznate stvari već istražuju i razmišljaju nad onim što govore, jedva da se dve slažu šta toj istoriji pripada i kako se zbivala. Već od početka: da li povest računara počinje kad je pronađen a) prvi hardver, b) prvi softver, c) prvi principi algoritamske nauke, ili možda d) prva ideja za njihovu masovnu upotrebu, ideja o personalcu? Dakle, da li je u početku bio a) abakus, b) Bebidž, c) Turing, d) Aj-bi-em? Svejedno vam je. Znači da osim onoga što vam je ispred nosa (večito neki monitor) pogled dopire još, evenutalno, do Vladine sklopke za slova. Dip-svič istorije je, verujte mi na reč, mnogo dublji.

Nisam pisac istorije, istoriograf, ali mogu zamisliti kako je uzbudljivo pisati istoriju na kompjuteru. Kad je arhiva na dohvatu dve-tri tipke, onda je ona već zagonetnija. Mnogi naši „istoričari" — i oni koji pišu istorije naroda, država, i oni koji pišu istorije književnosti, i oni koji pišu povest malih ljudi i malih stvari — gube pojavom kompjutera posao. Jedan uvaženi akademik je užasnut tom pojavom, razumem ga, jer ono što on mistifikujući mrči godinama — to računar obradi za nekoliko sati. Naprosto, zadatak se premestio u ozbiljnije polje i gaće su spale. Ah, mogu tek zamisliti koliko je uzbudljivo pisati istoriju samih računara, tj. cele te priče koja se okuplja oko soft-hardvera.

Pominjem istoriju jer mi se čini da će se, u povesti koja dolazi, ukrstiti dva „zanata" ili, bolje, dva tipa čoveka. Oni ionako mnogo čeg imaju zajedničkog, a od prepoznavanja i uzajamnog susreta može zavisiti sudbina komunikacionih mašina zvanih „računari". Pogađate, reč je o susretu Pisca, s jedne, i Hakera, s druge strane. Šta im je zajedničko? Pre svega retka vrsta radoznalosti u odnosu na tekstove. Haker barata *tekstovima* koliko i Pisac, njegovo čitanje ili pisanje „listinga" ne zaostaje za književnom meditacijom (jedino je nešto drukčije orijentisano). Potom, takođe retka vrsta ne-korisnosti, individualnosti, smisla za avanturu i larpurlartizam. Hakeri su umetnici bitova (za razliku od Tehnoprogramera, zanatlija bajtova). Ni njihovo carstvo nije nda ovoga sveta, a noćno bratstvo Pisaca i Hakera upućuje ih na savezništvo stvaranja (a ne stvarni savez). Kompjuteri neće ukinuti slučaj knjige, ali već pomišljam na sve one tekstove koji će se pisati da bi se čitali na monitoru — novele, priče, romane, drame, eseje gde će se od čitaoca tražiti da zaista učestvuje, i bira sebi odgovarajući zaplet, lik, rasplet. ... Nešto nalik na mobilni tekst, na Borhesa transkribovanog za monitor, par flopija i tvrdi desk.

Staviti pisca pred kompjuter — i upitati ga šta, kao pisac, misli o kompjuteru — znači dobiti neočekivane, pomalo fantastične odgovore. Počeo sam ovaj zapis (koji će se nastaviti negde drugde, koji se drugde mora nastaviti) fantazmama. Završiću ga jednim citatom: „Tu, gde stupismo u sferu matematika, sada smo unutar procesa koji su navodno najneljudskiji od svih ljudskih aktivnosti, i najudaljeniji od pesništva. I pak, upravo tu je sloboda mašte za umetnika najšira".

Reči su Hejvloka Elisa, psihoanalitičara, uzimam ih iz knjige Donalda Knuta, o metafontovima.

# POŠTA BEZ POŠTARA

**Kao što je 1981. godina donela bum džepnih računara, 1984. bum kućnih računara a 1987. godina bum personalaca, godina koju upravo privodimo kraju ostaće zapamćena kao godina komunikacija. Modema je sve više, računari se povezuju u mreže, komunikacija sa svetom je sve brža i jednostavnija, BBS-ovi doživljavaju procvat... Za većinu korisnika najinteresantniji vid komunikacija je elektronska pošta, medij koji je u razvijenim zemljama potpuno bacio u zapećak „klasična" pisma, koverte, marke pa čak i telefonske razgovore – izrazi *fax* i *e-mail* dobro su poznati čak i onima koji nikada nisu imali veze sa kompjuterima. Ova serija tekstova baviće se mogućnostima i korišćenjem elektronske pošte.**

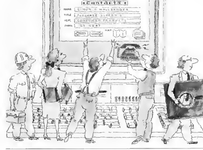
Dejan Ristanović

Kao što to često biva i u ovoj oblasti znanje često kaska za tehnologijom – zbog nedostatka dokumentacije, nepoznavanja jezika i, naravno, zauzetosti, mnogi korisnici koji imaju velike potrebe za elektronskom poštom i koji istovremeno imaju pun pristup nekom od sistema koji tu poštu podržava ne koriste njegove usluge ili ih koriste u minimalnoj meri, uz komfor koji je daleko ispod performansi njihovog sistema. U okviru ove serije tekstova obraćamo se relativnim početnicima u oblasti elektronske pošte koji bi želeli da uz minimum truda i troškova uspostave vezu sa svetom.

## SA VAX-OM NA TI

Elektronska pošta je postigla popularnost i zahvaljujući tome što je globalno prisutna – svaki ozbiljniji operativni sistem omogućava ne samo komunikaciju među korisnicima nego i razmenu poruka sa korisnicima nekog drugog (istorodnog ili potpuno nekompatibilnog) sistema koji je vezan u istu mrežu. Svaki operativni sistem, prirodno, poseduje svoj set komandi za rad sa porukama – na Unix-u se komunicira na jedan način, na VM-u na drugi a na VMS-u na treći. Princip je, srećom, dosta sličan tako da ćete, ako ste naučili da komunicirate sa jednog sistema, bez previše problema komunicirati i sa nekog sasvim različitog. Pošto nam prostor u „Računarima" omogućava da se posvetimo samo jednom operativnom sistemu, odlučili smo se za onaj koji je kod nas najrasprostranjeniji: zamislićemo korisnika koji sedi za svojim PC-jem koji je opremljen modemom i želi da komunicira sa svetom posredstvom nekog VAX minikompjutera koji radi pod operativnim sistemom VMS. Samim VAX-om i VMS-om bavili smo se ranije u „Računarima" (brojevi 40-44 našeg časopisa), tako da ćemo ovde opisivati samo one aspekte koji su bitni za elektronsku poštu.

Pošto se u ovom uvodnom tekstu upoznamo sa pozivanjem VAX-ova, osnovnim komandama VMS-a i onim što nam otvoreni radni nalog na nekom od sistema omogućava, u sledećim „Računarima" naučićemo kako se šalje elektronska pošta među računarima u zemlji. Zatim prelazimo na slanje pisama po svetu, komunikaciju među raznorodnim mrežama, prenos datoteka, usluge paketnih mreža, elektronski uređivane časopise koji pokrivaju najrazličitije oblasti kompjuterskog i nekompjuterskog života od viceva preko SF-a do nuklearnih elektrana (većinu ovih časopisa možete dobijati potpuno besplatno – dovoljno je da pošaljete poruku na odgovarajuću adresu i tekstovi potom sami stižu!), TRICKLE servere koji vam omogućavaju da bez ikakvih troškova dođete do megabajta i megabajta PD i *Shareware* programa za PC, Mac i druge računare... Biće zanimljivo!




Da biste koristili sve ove lepote potreban vam je pristup nekom od VAX-ova koji je povezan u svetsku paketnu mrežu. Pitanje kako da dobijete ovakav pristup (najvećim delom) izlazi iz okvira ove serije tekstova – možda će vam ga obezbediti vaša radna organizacija, možda fakultet koji pohađate, možda Univerzitet (dobar deo mašina vezanih u JUPAK pripada visokoškolskim ustanovama), možda ga dobijete kao VIP, možda ćete plaćati usluge (što ume da bude prilično papreno) itd. Sam pristup nekada nije dovoljan – neke od stvari o kojima ćemo govoriti su skupe i kao takve dopuštene samo određenoj grupi korisnika nekog računskog centra, za neke je možda potreban neki specijalni softver ili doinstalacija VMS-a i tome slično. Verujemo, međutim, da treba upoznati što više usluga koje sistemi elektronske pošte obezbeđuju – čak i ako vam neka od njih nije pristupačna, činjenica da ona postoji može vas inspirisati da je dokupite!

## USPOSTAVLJANJE VEZE

Počnimo dakle od početka – imate PC sa modemom i želite da komunicirate sa svetom. Preporučili bismo vam komunikacioni program *Procomm Plus* (punu a ne *Test Drive* verziju) koji treba da postavite na 2400 bps 8N1 i ne zaboravite VT102 umesto ANSI emulacije koju koristite za komunikaciju sa većinom BBS-ova. Jedna od veoma nezgodnih osobina VT102 emulacije jeste što je taster *Backspace*, koji na PC-ju služi za brisanje prethodnog karaktera, „preimenovan" u BOL (kurzor na početak reda) dok se za brisanje prethodnog znaka koristi Del (tačka na numeričkoj tastaturi). Ovo je konvencija koja će vas isterati iz kože – preporučujemo da komandom *Key Mapping* (Alt F8 – zato vam je potreban komercijalni a ne *Test Drive Procomm Plus*) promenite značenje tastera *Backspace* prepisujući pored njega ono što piše pored tastera *Del*. Ovo se, srećom, radi samo jednom – čim snimite promene na disk, spremni ste da „pričate" sa VAX-om!

Potrebno je, pre svega, da stupite u vezu sa računarom na kome vam je otvoren račun – na slici 1 dajemo nekoliko „važnih telefona" zajedno sa parametrima veze.

Šta da radite ako imate pristup nekom drugom sistemu koji nema modemske ulaze? Mo-

Slika 1



Slika 2



Slika 4



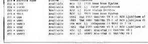
Slika 5



Slika 6



Slika 3



Slika 7



Slika 8



žele pozvati javni JUPAK broj ((011)235 1150 u Beogradu ili (041)235 244 u Zagrebu) a on da se, ako imate pristup JUPAK-u, priključili na željeni računar preko njegove mrežne adrese (NUA) – na primer da biste došli do čvora OLIMP (Institut „Ruđer Bošković" u Zagrebu) kucate Nime/lozinka-14140000309. Ako nemate pristup JUPAK-u, možete pozvati broj Univerziteta u Ljubljani pa koristiti njihov SET HOST servis (više o njemu docnije), ali je to posta nepogodno za nekoga ko bi, recimo, morao razgovarati sa sistemom koji mu je geografski veoma blizu preko udaljenog SET HOST čvora. Ukoliko ništa od ovoga ne pomaže, nema vam druge nego da umesto elektrona prenesete sebe: odete do računara na kome imate račun i radite „u lokalu" sa nekog od priključenih terminala.

Bilo kako bilo, po povezivanju sa sistemom najčešće ćete dobiti poruku terminal servera nalik na sliku 2.

Terminal server očekuje da otkucate vaše ime, ali je ovaj podatak sasvim irelevantan – zapravo, ono što otkucate nema nikakvog značaja pošto se radi samo o imenu pod kojim će se voditi ta seansa. Otkucajte dakle proizvoljnih par slova i sačekaće vas prompt Local> koji označava da terminal server očekuje vašu komandu. Sa HELP možete da dobijete podatke o svim raspoloživim komandama ali će vam u praksi biti potrebna samo jedna ili možda dve. Ona „manje potrebna" komanda je SHOW SERVICES, što može da proizvede ispis kao na slici 3.

Na ovaj način dobijate spisak svih uređaja sa kojima server može da vas poveže – imena su prilično kriptična pa pri dobijanju računa treba da se raspitate i kako se zove čvor na kome vam je račun dodeljen. Ukoliko je ime čvora UEK, otkucaćete CONNECT UEK i sačekati par sekundi da se veza uspostavi. Ukoliko uspostavljanje veze ne uspeva (računar je možda isključen) pritisnite Break (Alt B u *Procomm*-u) i, pošto se prompt Local> ponovo pojavi, pokušajte sa nekim drugim servisom.

Pošto je veza uspostavljena, pojavljuje se prompt Username i VAX očekuje da otkucate svoje korisničko ime. To je obavezno **jedna** reč od najviše petnaestak slova i brojeva (za potrebe elektronske pošte **vrlo** je preporučljivo da vaš *username* ima najviše 8 slova) koja je verovatno neka skraćenica od vašeg imena i prezimena, statusa i tome slično – svaki računski centar ima manje ili više razvijenu konvenciju o dodeljivanju korisničkih imena pa najčešće pročitavši nečije ime možete da zaključite da li se radi o običnom korisniku, nekome iz uprave sistema i tome slično. Korisničko ime je javna informacija i, kao takva, nije dovoljna za jednoznačnu identifikaciju korisnika. Zato će VAX, posle unetog imena, zahtevati i lozinku ispisujući prompt *Password*. Lozinka je proizvoljna reč koju znate samo vi i koju saopštavate samo sistemu – početna lozinka će vam biti dodeljena prilikom otvaranja računa, a docnije ste slobodni da je menjate prema svojim potrebama. Za razliku od lozinke, korisničko ime je uglavnom nepromenljivo (na njega može uticati samo Upravnik računskog centra) i pratiće vas kroz čitav život na izabranom VAX-u.

Pošto su korisničko ime i lozinka ispravno uneti, računar ispisuje pozdravne poruke i znak $ koji označava da je spreman za unos komandi. Na primer, kao na slici 4.

Smisao svakog od redova je uglavnom jasan: računar nas obaveštava kada smo se prethodni put interaktivno i neinteraktivno prijavili (interaktivno prijavljivanje je rad poput opisanog, neinteraktivno je slanje nekog posla na paketnu obradu ili komunikacija sa nekog dru-

gog VAX-a u mreži – više o tome docnije) što nam omogućava da proverimo da li je neko neovlašćeno koristio naš račun (događe se naročito onima koji biraju „neubedljive" lozinke) i prijavljuje da su elektronskom poštom pristigle 3 nove poruke koje treba da pročitamo

**Prve** stvar koju treba da uradite kada vam u nekom računskom centru otvore nalog jeste da promenite lozinku – ko zna gde je sve papirić na kome je lozinka pisala stajao! Treba otkucati set *password* uneti staru lozinku a zatim dva puta novu Prilikom izbora lozinke treba biti obazriv – **vrlo** je loša praksa davati lozinku koja predstavlja vaš nadimak ime nekog od članova porodice datum rođenja i tome slično – najbolje je da izmislite neki niz od 6-7 slova koja ne znače ništa Naravno da novu lozinku treba dobro zapamtiti – ako je zaboravite **niko** vam je ne može reći mada je Upravnik centra naravno može promeniti



## RAZGLEDANJE RAČUNA

Pošto ste se pobrinuli za bezbednost, treba da upoznate sistem i vaš novi račun Što se sistema tiče možete da otkucate *show users* i zateći će vas ispis poput slike 5

Važni elementi svakog od ovih ispisa su ime korisnika (*Username*) ime procesa (*Process Name*), broj procesa (PID) i oznaka terminala Imena korisnika su, sećamo se, fiksna i zadaje ih Upravnik centra pri otvaranju naloga Ime procesa korisnik samostalno bira ili je ono ako korisnik ništa ne preduzme što je ujedno i najčešći slučaj jednako korisničkom imenu PID je **jedinstveni** broj dodeljen vašem procesu dok vam oznaka terminala pomaže da saznate odakle se korisnik zapravo javlja – OPA0 je konzola oznake tipa TXA1 ili TXB5 označavaju neki od terminala koji se nalaze u samoj računarskoj sali oznake tipa LTA1234 označavaju pristup preko terminal servera odnosno često modema RTA1 označava pristup preko mreže dok se ostale „egzotične" oznake odnose na razne specijalne uređaje u sistemu

Sve ovo govorimo zbog situacije koja u praksi nije nipošto retka – veza se prekine ali VAX to ne shvati i vi ostajete neprekidno ulogovani Ukoliko ste pre prekida veze pozvali neki sistem u Americi ostaćete neprekidno na vezi satima i danima što vam baš neće pomoći da dobijete orden Zato u ovakvim situacijama **obavezno** treba ponovo pozvati prijaviti se otkucati *show users* i proveriti da situacija nije poput slike 6

Očigledno, postojala su dva procesa istog korisnika – obzirom da VMS ne dopušta dva istoimena procesa u jednoj grupi drugom je za ime automatski dodeljena oznaka terminala Prepoznali smo koji je proces onaj „mrtav" (ima manji broj terminala ono LTA473 umesto LTA474) i „ubili" ga navodeći PID procesa Tek kada je ova operacija završena možemo nastaviti sa radom ili se spokojno odjaviti komandom *logout*

Ponekada se desi da proces zapadne u neko čudno stanje hibernacije iz koga ga jednostavno ne možete ni probuditi ni zaustaviti – u takvom slučaju proverite da li se sigurno nalazite na istom čvoru mreže na kome je i taj proces i ako je to u redu potražite pomoć od operatera primenjujući komandu PHONE o kojoj ćemo tek govoriti

Pošto znate da se prijavite i odjavite vreme je da upoznate svoja prava Otkucajte show proc /all i dobićete nešto poput slike 7

Pokušajmo da identifikujemo neke od elemenata ove tabele Na početku je najvažnije vaše korisničko ime, identifikacioni broj procesa oznaka čvora i ime procesa Sledi ime terminala (mi smo bili prijavljeni preko mreže pa je terminal RTA1 logička oznaka UIC a (*User Identification Code* pogledajte našu seriju tekstova o VMS u koja je počela u Računarima 40 ) osnovni prioritet i ime tekućeg kataloga Pitanje osnovnog prioriteta nije jednostavno kako izgleda pošto VAX pri radu sa procesima postepeno povećava i umanjuje prioritete zavisno od zahteva koje pojedini procesi imaju ako je neki proces malo koristio procesor i dugo čekao dobija viši prioritet dok drugom procesu koji stalno „navaljuje" na procesor prioritet postepeno opada Vaš prioritet se obično kreće između baznog prioriteta i baznog prioriteta+5, bazni prioritet će za interaktivni rad obično (zavisno od organizacije računskog centra) biti 4 ili 5

Slede kvote vašeg procesa – ima tu svega i svačega i najčešće ne treba trošiti vreme na razumevanje pojedinosti radićete ono što vam je potrebno i ako sistem prijavi da ste prekoračili neku kvotu „kukati" kod Upravnika centra da vam je poveća Jedino što treba zapaziti je *CPU limit* odnosno ograničenje procesorskog vremena Elektronska pošta tipično troši veoma malo procesorskog vremena – događaće se da realno radite čitav sat i da potrošite svega 10-tak procesorskih sekundi Ukoliko međutim počnete da kompajlirate i linkujete ili da pretražujete neke baze podataka utrošak procesorskog vremena rapidno raste ako dostignete limit sistem vas nemilosrdno prekida što nije baš bezopasno po vaše podatke Oznaka *Infinite* označava da *CPU limit* za vas ne postoji

Poglavlje *Accounting information* takođe nije previše interesantno – podaci o tome koliko ste kojih resursa utrošili tokom seanse su od značaja jedino za račun koji će vam biti ispostavljen Svaki računski centar ima svoj način naplaćivanja usluga što znači da se svaka od stavki poglavlja *Accounting* ni ne mora naplaćivati

Sledi najzanimljiviji deo – *Process privileges* Na VAX-u je naime pitanje privilegija veoma detaljno razrađeno tako da je Upravniku centra omogućeno da precizno odredi koji korisnik može šta raditi Svaki korisnik će imati privilegiju TMPMBX (*may create temporary mailbox*) dok će oni kojima je dopušteno da opšte preko mreže i zovu druge čvorove imati i NETMBX (*may create network device*) Tu je ujedno i kraj što se „smrtnika" tiče – postoji čitav niz ostalih privilegija ali verovatno ni jednu od njih nećete videti Pomenućemo privilegije OPER (operator) BYPASS (mogućnost čitanja svih datoteka bez obzira na vlasnika) i najinteresantniju SETPRV koja vam omogućava da sebi dodelite bilo koju privilegiju koja vam je potrebna pa i da otkucate set *proc /priv=all* Počevši od verzije 4 VMS omogućava da imate i neke privilegije koje vam u startu nisu dodeljene tako da i ako nemate NETMBX možete pokušati set *proc /priv=netmbx* nadajući se da vam je ta privilegija dodeljena ali se ne nalazi na spisku početnih privilegija – davljenik se hvata i za slamku

Za razliku od oznaka privilegija koje je propisao DEC *Process rights identifiers* su manje-više lokalna stvar – svaki računski centar može deliti korisnike u klase i svakoj klasi dopuštati pristup nekim opcijama sistema Imena tih klasa su proizvoljna tako da tumačenje spiska nije uvek jednostavno – za sada zapamtite da pominjanje reči PSI *verovatno* označava da se sa vašeg računa mogu zvati drugi sistemi po svetu (to poprilično košta računski centar!) dok EARN označava mogućnost slanja elektronske pošte u inostranstvo Ostatak spiska su podaci od vrlo malog značaja čak i za iskusnog korisnika VMS-a

U upoznavanje svog računa spada i komanda *show quota* koja će proizvesti nešto poput slike 8

Radi se o maksimalnom prostoru na disku koji smete da zauzmete – blok ima 512 bajtova što znači da korisnik kome je dodeljeno 2000 blokova može da zauzme megabajt diska Dopušteno prekoračenje označava da će vam sistem kada ste prešli kvotu dozvoliti malo više prostora zbog izlaska iz editora i sličnih operacija Ukoliko međutim primetite da vam se kvota smanjuje svakako obrišite neke datoteke (*purge* je dobar prvi korak) kako docnije ne biste zapali u nevolje

Alternativno komanda *show quota* može da bude kao na slici 9 što praktično znači da na datom disku nema ograničenja u zauzimanju prostora

U sledećim „Računarima" upoznaćemo još neke komande VMS a i početi da šaljemo poruke

*Zahvaljujemo se EARN u (Beograd) institutu Informatičkih Znanosti iz Maribora (RCUM) Računskom centru Elektrotehničkog fakulteta u Beogradu (BUEF) i Univerzitetu u Ljubljani (UEK) na pomoći pri realizaciji ove serije tekstova Komentare očekujemo u Sezamovoj konferenciji KOMUNIKACIJE ili u vidu privatnih poruka autoru (mail send dejanr) Takođe preko DECnet a na adrese UBBG ERISTAN UEK RACUNARI ili RCUM RACUNARI*

# PROZORI NA KLIPERU

***Clipper* je, pre svega, programski jezik za obradu podataka. To, međutim, ne znači da je za druge važne funkcije jednog korisničkog paketa, kao što je takozvani „korisnički interfejs", potrebno posezati za nekim drugim programskim paketom. *Clipper* je opremljen skupom funkcija koje omogućuju – mogli bismo da kažemo jednostavnu – izgradnju kompletnog korisničkog interfejsa.**

Dalibor Pujić

Komunikacija jednog programa sa svojim korisnikom odvija se kroz

## ČETIRI SEGMENTA

Prvi segment obuhvata **ponudu za izbor operacije/funkcije**, odnosno grupe operacija/funkcija programa koje korisnik može da koristi u određenom delu programa. On ne isključuje mogućnost prikaza i ostalih funkcija koje u tom trenutku nisu dostupne, a od njegove jasnoće zavisi i jednostavnost korišćenja programa (širina kruga potencijalnih korisnika iz grupe kojoj je program namenjen, dužina i težina obuke za njegovo korišćenje i sl.). U ovom segmentu iskusni programeri se koriste provereno tehnikom prepoznatljivosti, odnosno usvajanjem principa široko rasprostranjenih programskih paketa, tako da korisnik prepoznajući u programu sistematiku drugih programa, a poznajući predmet rada, lako i bez naročite obuke uspešno koristi program.

Drugi segment obuhvata **ulazne panele** pomoću kojih korisnik unosi podatke u svoje datoteke ili ih preko njih menja. I u ovde je korisno ići na sistem prepoznatljivosti. Naime, retko se događa da se pomoću računara rade potpuno novi poslovi, poslovi koji ranije nisu rađeni „pešake". U takvom radu koriste se najrazličitiji tipovi obrazaca, formulare i sličnih papire. Njihov izgled, uz malo mašte, lako je preneti na ekran, a to će zasigurno smanjiti „šok" korisnika u susretu sa novom tehnologijom i u mnogome ubrzati njegovo „uigravanje" u novim uslovima.

Treći segment obuhvata **upozorenje na grešku**. Uz svu logičku kontrolu koju programeri obezbeduju, nemoguće je otkloniti mogućnost da korisnik načini grešku koju će računar brzo konstatovati i na nju odgovoriti prestankom rada ili blokiranjem, što je svakako nepoželjno. U takvim slučajevima, poželjno je korisniku obezbediti informaciju gde je i šta pogrešio, kao i konforan povratak u rad sa programom.

Četvrti segment obuhvata **izlazne izveštaje** kroz koje korisnik preko ekrana ili štampača dolazi do rezultata koji su mu potrebni i zbog kojih, na kraju krajeva, i prikuplja i unosi podatke. Bez obzira da li se radi o prostom ispisivanju rezultata ili njihovim kompleksnijim analizama i statističkoj obradi, ovaj segment treba uz klasične forme da obezbedi i čitav niz novih izveštaja. Pošto je u potpunosti zavisan od predmeta rada, o ovom segmentu više neće biti reči u ovom tekstu, ali će vam malo mašte i truda sigurno omogućiti da iz prethodnih prenesete neka rešenja i u ovaj segment, posebno kad se radi o ekranskim izveštajima.

## IZRADA MENIJA

### CRTANJE ekranskih prikaza

Vrlo jednostavan, očigledan i efikasan metod za izradu ekranskih prikaza je korišćenje TEXT komande i ASCII karaktera. Očigledan i jednostavan zato što već u toku pisanja programa možete da vidite izgled budućeg ekranskog prikaza, a efikasan zato što dobro i brzo funkcioniše.

Kako to treba uraditi?

- Posle TEXT komande u programu ostavite 23 prazne programske linije i dajte komandu ENDTEXT.
- Kursorom podesite izgled ekrana u editoru tako da pred sobom imate celu praznu površinu između TEXT i ENDTEXT komandi.
- Kursor locirajte na željeni početak (jedan od čoškova „slike" ili bilo koja druga, vama odgovarajuća pozicija).
- Korišćenjem ALT tastera i numeričke tastature unesite grafičke simbole koji će oblikovati i finalizirati vaš ekran, a unutar formirane slike unesite adekvatan tekst.
- Ovo ćete izvesti mnogo jednostavnije ako pred sobom imate tabelu ASCII vrednosti grafičkih simbola.

```
SINTAKSA: @ 12,41,22,70 BOX chr(218)+chr(196)+chr(191)+...+chr(176)
ili
@ 12,41,22,70 BOX izgled
```

Slika 1

```
- izgled = chr(218)+chr(196)+chr(191)+chr(179)+,
       chr(217)+chr(196)+chr(192)+chr(179)+,
       chr(176)
```

Slika 2

```
- @ 5,5 TO 10,10           daje pravougaonik uokviren jednom linijom
- @ 5,5 TO 10,10 DOUBLE    daje pravougaonik uokviren dvema linijama
```

Slika 3

Šta treba imati u vidu pri korišćenju ovog metoda?

- Prvo, neophodno je da se u programu neposredno pre upotrebe TEXT komande ekran obriše (CLEAR). Izdavanje ove komande će sadržaj TEXT-a ispisati na ekran. Ako ste radili onako kako je sugerisano, sa ostavljanjem 23 linije između komandi, vaš novi tekst će istisnuti prethodni sadržaj ekrana i neće biti naročitih problema. Ipak, ako smanjite razmak, što je i logično nekon završetka slike, vrh ekrana će ostati ispisan. Osim toga, CLEAR treba da vas upozori da sačuvate/snimite prethodni ekran ako vam je negde ponovo potreban.
- Takođe, to je samo tekst, „mrtvo slovo na ekranu" koje tek treba programski oživeti. Za to su vam potrebne pozicije teksta koji želite da „oživite" kroz meni, da bi ih kasnije koristili u komandi MENU TO.
- Posebnu pažnju posvetite redosledu ispisanog teksta koji hoćete da koristite kao ponudu (isti taj redosled, kasnije treba preneti u MENU TO.
- U kreiranju grafičkog izgleda ekrana (jednostruki i dvostruki ramovi i ostale geometrijske vratolomije) planirajte mesto za poruke koje će pratiti ponudu (komanda PROMPT/MESSAGE) i koje će vam uvek biti u istoj liniji ekrana.

## POZICIONIRANJE

Takođe efikasan, ali ne toliko očigledan metod izrade ekranskih prikaza, je korišćenje grupe komandi za ekranski ispis (BOX, SAY/GET, ?, ??, REPLICATE, DOUBLE...). On je nešto komplikovaniji zato što je korišćenje pojedinih funkcija uslovljeno njihovim prethodnim definisanjem (pre upotrebe), a za pojedine vrste grafičkih prikaza neophodna je kraća ili duža programska procedura, u svakom slučaju, bar jedna programska linija.

U korišćenju ovog metoda treba imati na umu da svaka komanda pokriva onu prethodnu u slučaju da im se pozicije delimično ili potpuno preklapaju, zato ih ili razdvojte ili prilagodite za zajedničko delovanje (ako se dodiruju, može se nešto i postići).

BOX – komanda kojom se crta četvorougaoni okvir definisanog izgleda i veličine. Izgled se definiše pomoću 9 parametara koje možete zadati u izvornom izgledu („–"), ili preko ASCII vrednosti (CHR(188)). Prvih 8 od tih devet parametara definiše izgled okvira vašeg četvrorougaonika (1 gornje levo teme, 2 gornja linija, 3 gornje desno teme, 4 desna bočna linija... itd. u smeru kretanja skazaljke na časovniku), a deveti parametar definiše podlogu unutar četvorougaonika. Ako deveti parametar nije definisan/naveden, podloga će biti ista kao i na ostalim delovima ekrana van vašeg boksa. Veličina se definiše pomoću 4 parametra (1 gore levo red, 2 gore levo kolona, 3 dole desno red, 4 dole desno kolona) kao na slici 1.

Za korišćenje BOX-a po drugom modelu potrebna je prethodna definicija „izgleda" kao na slici 2.

## Listing 1

```
SET CURSOR OFF
SET TALK OFF
ulaz=
RAM1=CHR(220)+CHR(176)+CHR(181)+CHR(188)+CHR(018)+CHR(180)+CHR( V  CHR (180)
RAM2=CHR(200)+CHR(200)+CHR(180)+CHR(180)+CHR(180)+CHR(200)+CHR(270)+CHR(176)
RAM3=CHR(200)+CHR(205)+CHR(180)+CHR(185)+CHR(170)+CHR(200)+CHR(011 +CHR(170)
RAM4=CHR(220 +CHR(176)+CHR(200)+CHR(180)+CHR(170)+CHR(070)+CHR(011 +CHR(190)
boja_1="W/B,B/W  ,W/N"
boja_2="W/N N/B   N/W"
W"          "
CLEAR
@ 1 2,20 22 BOX RAM1
@ 8 8 20 70 BOX RAM2
@ 5 0 20 20 BOX RAM3
@ 0 27 8 86 BOX RAM3
@ 8 8F SPE  BLOCE I KASETE
@ 9 15 SAY  Program za sredjivanje i vodjenje podataka o "
@ 20 20 SAY "diskutakama i kasetotakama i stampu kataloga"
@ 22 9 18 87 BOX RAM3
@ 20 88 SAY "D P P"
@ 16 88 SAY  dodige 6 andiwear"
@ 88 89 PPE "  Blaska Perovic - Hedrikanca 88              (c) PED-PPE
SETCOLOR(BOJF_8)
@ 20 20 say  your code " GPI 0
READ
CLEAR
SETCOLOR(BOJF_1)
IF UPPER(D)="(SPRA"
JA=F
DO WHILE RA=0
CLEAR
  @ 0 0 8 88 BOX RAM3
  @ 2F,8 88 8 BOX RAM3
  @ 20 1 CLEAR TO 28 P
  @ 81 8,26 8 BOX RAM4
  @ 82 2 PPE  D P P"
  @ 82 8 PPE  "GE"
  @ 8 20 SAY  PLOCE I KASETE
  @ 1 28 2 88 BOX RAM3
  @ 8 12 SAY  Program za vodjenje i sredjivanje diskotekE i kasetoteke"
  @ 2,1 10 8P BOX RAM3
  @ 20 20 SAY "KRATKO UPUTSTVO:
  @ 22 P SAY "Izbor zeljenog podrucja vrsi se dovodjenjem inverzne povrsine na
  @ 22 P SPE "zeljeno polje (pomocu kursor strelica) i pritiskom na ENTER taster
  @ 20 P SAY "Tekst POMOCA daje Vam opsti opis izbora koji je ponudjen
  @ 86 P SAY "Tekst AKCIJE daje sta ce konkretno uraditi ako se izbor izvrsi
  SET MESSAGE TO 8 CENTER
  @ 8 8 P 22 BOX RAM3
  @ 8 2 SPE  POMOCA :
  SET KRAJ ON
  @ 22 8 PROMPT  U N P P  MESSAGE  UNOS PODATAKA
  @ 12 88 PROMPT "IPMENE  MESSAGE  IPMENE PODATAKA
  @ 12 85 PROMPT "STAMPP  MESSAGE "STAMPANJE KATALOGA I OMOTP
  @ 18 65 PROMPT "P R P J  MESSAGE "KRAJ PROGRAMA I POVRATAK U DOS
  @ 11 8 87 27 BOX RAM3
  MENU TO IZBOR
  DO CASE
*********************     UNOS  ************************
    CASE IZBOR=1
UN=0
DO WHILE UN=0
  @ 10 2 SPE  AKCIJE   "
  SET MESSAGE TO 88 CENTER
  @ 22 2 18 13 BOX RAM3
  @ 82 8 PROMPT " KASETE  " MESSAGE "UNOS PODATAKA NOVE KASETE"
  @ 18 8 PROMPT "PLOCE" MESSAGE "UNOS PODATAKA NOVE PLOCE
  @ 15 5 PROMPT  KRAJ  MESSAGE "KRAJ UNOSA I POVRATAK U OSNOVNI MENI
  MENU TO UNOS
    DO CASE
***********************   UNOS KASETE
        CASE UNOS=1
***********************   UNOS PLOCE
        CASE UNOS=2
***********************   NOVA RAM3
        CASE UNOS=3
          UN=1
ENDCASE
ENDDO
  *******************    I Z M E N E  ******************
    CASE IZBOR=2
IZ=0
DO WHILE IZ=0
  @ 88 2 SAY "AKCIJE
  SET MESSAGE TO IF CENTER
  @ 18 20 16 88 BOX RAM3
  @ 18 28 PROMPT " KASETE  MESSAGE  IZMENA PODATAKA NEKE KASETE"
  @ 18 85 PROMPT  PLOCE  MESSAGE "IZMENA PODATAKA NEKE PLOCE
  @ 15 25 PROMPT  KRAJ  MESSAGE "KRAJ IZMENE I POVRATAK U OSNOVNI MENI
  MENU TO IZMENA
    DO CASE
************************   IZMENA KASETE
        CASE IZMENA=1
************************   IZMENA PLOCE
        CASE IZMENA=2
************************   IZMENA KRAJ
        CASE IZMENA=3
          IZ=1
    ENDCASE
    ENDDO
  ******************    S T A M P A  ****************
CASE IZBOR=3
ST=0
DO WHILE ST=0
  @ 17,7 SAY "AKCIJE : "
  SET MESSAGE TO 17 CENTER
  @ 12 82 17 87 BOX RAM3
  @ 12 88 PROMPT  KASETE/KAT  " MESSAGE  STAMPA KATALOGA KASETA
  @ 18 85 PROMPT "KASETE/OM  MESSAGE  STAMPA OMOTA ZA KASETU
  @ 15 85 PROMPT "PLOCE/KAT  MESSAGE "STAMPA KATALOGA PLOCA"
  @ 14 88 PROMPT  K R A J" MESSAGE "KRAJ STAMPE I POVRATAK U OSNOVNI MENI"
  MENU TO STAMPA
  DO CASE
  *********************   STAMPA KATALOGA KASETA
    CASE STAMPA=1
  *********************   STAMPA OMOTA ZA KASETE
    CASE STAMPA=2
  *********************   STAMPA KATALOGA PLOCA
    CASE STAMPA=3
  *********************   STAMPA KRAJ
    CASE STAMPA=4
      ST=1
      CLEAR
ENDCASE
ENDDO
***********************    K R A J  *********************
   CASE IZBOR=4
KR=0
DO WHILE KR=0
  @ 17 7 SAY "AKCIJE
  SET MESSAGE TO 19 CENTER
  @ 18 68 85 21 BOX RAM3
  @ 12 68 PROMPT   DOS    MESSAGE "KRAJ PROGRAMA I POVRATAK U DOS"
  @ 18 88 PROMPT "MENU  MESSAGE "POVRATAK U OSNOVNI MENI"
  MENU TO KRAJ
    DO CASE
      CASE KRAJ=1
        KR=1
        JA=1
        CLEAR
      CASE KRAJ=2
      KR=1
      CLEAR
  ENDCASE
ENDDO
  ENDCASE
ENDDO
ENDIF
IF UPPER(Q) (SPRA
   CLEAR
   @ 18 88 SPE "  "
   ?    Identifikacija nije u redu  nemoguca upotreba!!!"
   ?
   ?
   ?
   ?
ENDIF
QUIT
```

Meni u stilu programa *WordStar*

## Listing 2

```
CLEAR
SET TALK OFF
SET SCOREBOARD OFF
SET CURSOR OFF
SET KEY 88 TO POMOCA
  RAM =CHR(88F)+CHR(186)+CHR(181)+CHR(18P)+CHR(028)+CHR(198)+CHR(192)+CHR(2F9)
  RAM8=CHR(8T1)+CHR(8T8)+CHR(188 +CHR(186)+CHR(168)+CHR(8F8)+CHR(8T8 +CHR(208)
  RAM8+CHR(818)+CHR(888)+CHR(186)+CHR(188)+CHR(198)+CHR(8F8)+CHR(888)+CHR(889)
  RAM4+CHR(288)+CHR(882+CHR(1F8)+CHR(1F6)+CHR(198)+CHR(8F8)+CHR(811)+CHR(88F)
  boja_1="W/N N/W   N/W"
  boja_2="W+/P,N/N   N/W"
  boja_3="W+/N,N/N   N/N"
  boja_8="W/N N/N   N/8"
N=" "
SETCOLOR(BOJF_8)
  @ 0 8F SAY "PLOCE I KASETE   program za pracenje sadrzaja diskoteke/kasetoteke"
  ? "           "+REPLICATE("*  80)
  @ 0 2 2 8 BOX RAM8
  @ 1 2 SPE  D P P
  @ 2 2 SAY  dsa"
SETCOLOR(BOJF_2)
  KN=1
  DO WHILE KN=1
    @ 21 8 CLEAR TO 28 28
    @ 81,20 say "
    wait
  ENDDO
PROCEDURE POMOCA
SETCOLOR(BOJA_2)
@ 21 19 SAY  Unite pocadu "
@ 22 19 SAY  ┌──────────────────────────────┐
@ 22 19 SAY  │ Due     Izmena    Stampa    KA│
@ 28 1P SAY  └──────────────────────────────┘
SETCOLOR(BOJA_8)
@ 88 88 SAY "U
@ 28 88 PPE "I"
@ 88 81 PPE "8"
@ 88 8T PPE "8"
SETCOLOR(BOJF_2)
@ 81 88 GET M PICTURE
READ
SETCOLOR(BOJF_1)
@ 88 P CLEAR TO 28 88
DO CASE
CLEAR
   CASE M="U
     @ 88 P CLEAR TO 86 86
     @ 81 17 SPE "unos podataka "
     @ 22 17 SPE "┌────────────────────┐
     @ 22 19 SPE  │ Kasete    Ploce  Kraj│
     @ 88 19 SAY  └────────────────────┘
     SETCOLOR(BOJA_8
     @ 88 18 SPE "P"
     @ 88 88 SPE "P"
     @ 28 21 SPE  K"
     SETCOLOR(BOJA_2)
     D="
```

Program se nastavlja na sledećoj strani

## Nastavak listinga 2

```
@ 12,12 GET P PICTURE "!"
   READ
   SETCOLOR(BOJA_1)
   DO CASE
      CASE P="S"
      CASE P="E"
      CASE P="A"
   ENDCASE
 CASE M="1"
   @ 21,8 CLEAR TO 28,78
   @ 21,12 SAY "Izmena podataka "
   @ 22,10 SAY "┌──────────────────────────┐"
   @ 22,10 SAY "│  Isteto    Ioco    k aj  │"
   @ 24,10 SAY "└──────────────────────────┘"
   SETCOLOR(BOJA_3)
   @ 22,12 SAY "S"
   @ 22,12 SAY "E"
   @ 23,21 SAY "K"
   SETCOLOR(BOJA_2)
   P=" "
@ 21,12 GET P PICTURE "!"
   READ
   SETCOLOR(BOJA_1)
   DO CASE
      CASE P="S"
      CASE P="E"
      CASE P="R"
   ENDCASE
 CASE M="S"
   @ 21,8 CLEAR TO 28,78
   @ 21,22 SAY "Stampa"
   @ 22,10 SAY "┌────────────────────────────────────────┐"
   @ 22,10 SAY "│ ssete/EKT  Ioco/EKT  k este/Smet  k aj │"
   @ 24,10 SAY "└────────────────────────────────────────┘"
   SETCOLOR(BOJA_2)
   @ 22,12 SAY "E"
   @ 22,20 SAY "S"
   @ 22,27 SAY "A"
   @ 22,00 SAY "A"
   SETCOLOR(BOJA_2)
   P=" "
@ 21,12 GET P PICTURE "!"
   READ
   SETCOLOR(BOJA_1)
   DO CASE
      CASE P="E"
      CASE P="E"
      CASE P="E"
   ENDCASE
 CASE M="K"
   @ 21,8 CLEAR TO 28,78
   WAIT "             DA LI STE SIGURNI (D/N) " TO SANE
   IF UPPER(SANE)="D"
      KR=2
      CLEAR
   ENDIF
ENDCASE
@ 21,8 CLEAR TO 28,78
RETURN
```

Meni u stilu *Word*

## Listing 3

```
CLEAR
SET TALK OFF
SET SCOREBOARD OFF
SET CURSOR OFF
SET KEY -2 TO UK
SET KEY -3 TO UP
SET KEY  8 TO IS
SET KEY -2 TO IS
SET KEY -5 TO SRB
SET KEY -7 TO SRO
SET KEY -8 TO SPR
SET KEY -9 TO KRAJ
```

```
RAM1=CHR(218)+CHR(196)+CHR(191)+CHR(179)+CHR(217)+CHR(196)+CHR(192)+CHR(179)
RAM2=CHR(201)+CHR(205)+CHR(187)+CHR(186)+CHR(188)+CHR(205)+CHR(200)+CHR(186)
RAM3=CHR(212)+CHR(202)+CHR(184)+CHR(179)+CHR(190)+CHR(205)+CHR(212)+CHR(179)
RAM4=CHR(218)+CHR(196)+CHR(183)+CHR(186)+CHR(188)+CHR(180)+CHR(211)+CHR(186)
boja_1="W/B,B/W,,,B/W"
boja_3="W+/B,B/W,,,W/B"
boja_8="B/B,W/B,,,W/B"
boja_2="W/B,B/W,,,B/W"
SETCOLOR(BOJA_8)
@ 0,0 CLEAR TO 2,22

@ 0,2,5,10 BOX RAM4
@ 0,8 SAY "UNOS "
@ 1,8 SAY " PODATAKA"
@ 2,2 SAY "F2  KASETE"
@ 8,2 SAY "F4 - PLOCE"

@ 0,10,5,22 BOX RAM4
@ 0,21 SAY " IZMENA"
@ 1,21 SAY " PODATAKA"
@ 2,20 SAY "F5 - KASETE"
@ 8,20 SAY "F6  PLOCE"

@ 0,28,2,20 BOX RAM4
@ 0,28 SAY " STAMPA PODATAKA "
@ 2,27 SAY "F7 - KASETE ONO"
@ 2,27 SAY "F8 - KASETE KATALOG"
@ 8,27 SAY "F9 - PLOCE KATALOG"

@ 1,02,8,72 BOX RAM4
@ 2,00 SAY " KRAJ"
@ 2,00 SAY " EIO"

M=" "
SETCOLOR(BOJE_2)
  KR=1
  DO WHILE KR=1
     @ 10,10 say " "
     wait " "
  ENDDO
PROCEDURE UK
   @ 10,10 SAY "RAD1 UK"
   WAIT " "
   @ 10,10 SAY "                              "
RETURN
PROCEDURE UP
   @ 10,10 SAY "RAD1 UP"
   WAIT " "
   @ 10,10 SAY "                              "
RETURN
PROCEDURE IS
   @ 10,10 SAY "RAD1 IS"
   WAIT " "
   @ 10,10 SAY "                              "
RETURN
PROCEDURE IS
   @ 10,10 SAY "RAD1 IS"
   WAIT " "
   @ 10,10 SAY "                              "
RETURN
PROCEDURE SRB
   @ 10,10 SAY "RAD1 SRB"
   WAIT " "
   @ 10,10 SAY "                              "
RETURN
PROCEDURE SRO
   @ 10,10 SAY "RAD1 SRO"
   WAIT " "
   @ 10,10 SAY "                              "
RETURN
PROCEDURE SPR
   @ 10,10 SAY "RAD1 SPR"
   WAIT " "
   @ 10,10 SAY "                              "
RETURN
PROCEDURE KRAJ
   @ 10,10 SAY "RAD1 KRAJ"
   WAIT " "
   @ 10,10 SAY "                              "
   CLEAR
   KR=2
RETURN
```

Meni u stilu programa *dBASE*

Ako već koristite BOX preporučljivo je prethodno na početku programa definisati izgled svih potencijalnih BOX-ova kao niz parametara a kroz program kasnije pozivati samo naziv niza (drugi model). Takođe je preporučljivo, osim ako ne znate šta da radite sa vremenom da ne pravite BOX-ove koji već postoje kroz komande za crtanje četvorougaonika uokvirenih jednom ili dvema linijama (@ m,n TO i,j /DOUBLE) ili praznog prostora (@ m,n CLEAR TO i,j).

Kako je već napomenuto „crtanje" po ekranu se može izvesti i korišćenjem @ m,n TO i,j (DOUBLE/CLEAR). Konkretno rešenje daje slika 3.

Ipak, za razliku od BOX-a, unutrašnjost ovih pravougaonika se neće izmeniti (ako ste imali na tom delu ekrana neki tekst ili sl. on se neće izgubiti jer ovde nemate mogućnost definisanja podloge unutrašnjosti pravougaoni-

```
@ 5,5 SAY   "   " gornja linija četvorougaonika
A=6 DO WHILE A<10
@ a,5 SAY "          "   bočne linije četvorougaonika
A=A+1 ENDDO
@ 10,5 SAY   ""    donja linija četvorougaonika
```

**Slika 4**

ka). Da bi to potpuno funkcionisalo upotrebićemo @ 5,5 CLEAR TO 10,10. Time se efekat izjednačuje sa BOX.

Ako vam je ovo prejednostavno a vreme vam nije problem, postoji i sledeće rešenje (?!). Ovaj programčić (slika 4) će ostvariti isti efekat na ekranu.

Ako imate vremena na pretek i volite dugačke i programe opterećene gomilom izlišnih linija, sve ovo možete uraditi i pomoću ?, ? SPACE(n), ? REPLICATE(CHR(n),m). Nadam se da ne očekujete i taj programčić, ali evo sintakse i kratkog opisa ovih komandi:

- ? – ispisaće prazan red od pozicije kursora
- ? „tekst" – ispisaće tekst od pozicije kursora
- ? SPACE(20) – ispisaće niz od 20 blanko karaktera
- ? REPLICATE(CHR(196),20) – ispisaće niz od 20 karaktera 196 (─)

Ovu grupa komandi se izvršava neposredno iza pozicije kursora i pokriva samo onaj deo

ekrana koji je potreban i dovoljan za njihov ispis. Znači, pazite šta sve treba prethodno i naknadno izbrisati, a naročito gde vam se kursor u momentu upotrebe ovih komandi nalazi!

## "OŽIVLJAVANJE" EKRANA

Koliko god to izgledalo možda i banalno, preporučljivo je svakom programu dati pristup ni ekran ili uslovno nazvano „korice programa" kojim ćemo obezbediti predstavljanje programa (njegovog autora, namenu i verziju i sl.) ali i sprečiti neovlašćen pristup. Znači, uz crteže i ispis teksta, moraćemo da obezbedimo i prihvat (unos) koda za identifikaciju (jedan predlog mogućeg rešenja nalazi se na početku prvog programskom primera – dBASE meni). Ako neko ne želi ništa više od ekranskog ispisa, znači samo transparent na kome će nešto pisati (ovo je moguće koristiti u programu za slanje kratkih informativnih poruka), može se koristiti INKEY() komandom koja će nekoliko sekundi (ili koliko već želite) zadržati sliku na ekranu.

U ovom smislu, postoje dve varijante INKEY funkcije:

- INKEY(0) kada je program čeka do prvog pritiska na neki od tastera (isto kao i WAIT ali bez poruke)
- INKEY(n) kada program čeka n sekundi ili do pritiska bilo kog tastera.

Ipak, „pravi" meni zadaće nam malo više posla. Za ovu priliku napravićemo „koktel", odnosno nekoliko varijanti nastalih na osnovu prepoznatljivih osnovnih menija. Ja sam izabrao tri programska paketa kao uzorke za koje verujem da su kod nas u velikoj upotrebi i da svaki korisnik poznaje bar jedan od njih. To su *dBASE*, *WS* i *MS Word*. Da bi u našem meniju korisnik lako prepoznao principe jednog od ovih paketa, opisaćemo ih pre njihove izrade na najopštiji način, i to samo u delu koji se odnosi na meni.

### Meni a la *dBASE*

Kao osnovne karakteristike dB menija možemo navesti: pri vrhu ekrana, ispod naziva programa i još nekoliko eventualnih poruka (ali nikako ne o opisu ponude), nalazi se u horizontalnom nizu ponuda opštih opcija programa (u našem slučaju unos, izmena, izveštaji i prestanak). Izbor se vrši pomoću kursorskih strelica dovođenjem inverzne površine (izgled negativa) na opciju koja nas zanima i pritiskom na ENTER (CR/RETURN) taster ili pritiskom slova na tastaturi kojim počinje naziv željene opcije. Nakon izbora opšte ponude dobijaju se segmenti unutar te ponude koji predstavljaju jednu od konkretnih operacija (u našem slučaju, recimo kod izbora unosa, to su: unos novih ploča, unos novih kaseta i povratak u osnovni meni). Metod izbora se ponavlja, znači isti je kao i kod osnovne ponude. Takođe, za ovaj sistem bitne su poruke koje prate ponudu. Šta to znači? Ako izbor vršite pomoću kursorskih strelica, promena inverzne poršine (bez ovete, odnosno definitivnog odabira) dovodi i do promene sadržaja teksta koji opisuje ponudu. U našem slučaju, znači, imaćemo dve vrste opisnih poruka: za opštu i konkretnu ponudu. Ove poruke moraju da budu na ekranu dok god izbor traje (sve dok se definitivno ne završi). I još samo jedna važna karakteristika: Radni prostor, odnosno površina ekrana na kojoj se odvija vaša izabrana opcija programa nalazi se na posebnom ekranu. To znači da slika menija nestaje kad se pređe na konkretan rad, a vraća se nakon njegovog završetka i to onakva kakva je bila pre početka tog rada (znači da će izgled ekrana morati da se sačuva).

Koliko god to izgleda komplikovano, u izradi je mnogo jednostavnije i izgleda kao u programskom primeru na listingu 1.

### Meni a la *Word*

Specifičnosti ovog sistema rada sa programom/programskim paketom ogledaju se u sledećem: na ekranu se nalazi samo okvir (ram) radnog prostora koji tek nakon izbora opcije menja svoj izgled. Da bi se došlo do prilike za izbor opcije, mora se pritisnuti ESC taster, a nakon toga dobija se opšta ponuda. Izborom unutar opšte ponude, na njenom mestu pojavljuje se konkretizovanija ponuda. Nakon tog izbora ponuda nestaje sa ekrana i ponovo ostaje samo radni prostor. Ponuda se nalazi u samom dnu ekrana (zadnja tri reda), a izbor se vrši izborom istaknutog slova (tekst ponude se piše malim slovima osim jednog slova koje je veliko i čijim pritiskom na tastaturi vršite izbor). Ovde nema čuvanja ekrana (mada vam to niko ne brani), a grafičke egzibicije tipa ramovi i slično su bitno skromnije, a nema ni pratećih opisnih poruka (primer na listingu 2).

### Meni a la *WordStar*

U *WS*-u ekran je podeljen na dva dela. Gornja trećina je panel sa izborom opcija, a donje dve trećine su radni prostor. U *WS*-u nema šetanja kursora, sve se odvija pritiskom na označeni taster (kombinaciju tastera) koji su istaknuti u ponudi. *WS* ima veliki broj opcija i zbog toga postoji više specificiranih panela u zavisnosti od segmenta rada. U našem slučaju jedan će biti sasvim dovoljan i on će sve vreme biti na ekranu. Na samom dnu *WS* ekrana nalaze se ispisane funkcije 10 PF (funkcijskih tastera), ali je sistem identičan kao i kod gornjeg panela. Pritisak na taster proizvodi željeni rezultat. Vi ćete svakako dopuniti svoj program i tom varijantom, ako uspete da smislite šta da dodelite kao funkciju tim tasterima (programski primer na listingu 3).

Za korišćenje ovog ili sličnih sistema, neophodna vam je tabela ASCII vrednosti pojedinih tastera ili njihovih kombinacija (sa ALT, CTRL ili SHIFT tasterima).

## ULAZNI PANELI

Kako je već i napomenulo, ovo je značajan segment koji će bitno uticati na kvalitet i upotrebljivost vašeg programa. Ne misli se pri tom samo na prepoznatljivost izgleda ekrana/ulaza. Ovo je pravi trenutak za logičku kontrolu, ažuriranje podataka, moguće ažuriranje izveštaja i slično. O svim ovim i gomili sličnih činilaca morate da vodite računa, jer kako ste posejali... Ipak, to je stvar koju treba da sami uradite. U ovom tekstu biće reči o tome kako da lako i brzo organizujete osnovu kroz koju ćete kasnije realizovati i sve ostale zahteve koje pred sebe i program postavite.

Najgrublja podela ovog segmenta mogla bi da se izvrši na informativni i „pravi" unos. Ovo su naravno uslovno izabrani termini, a predstavljaju informativni – počev od identifikacije na početku programa pa sve dalje do kraja, preko pitanja tipa „Hoćete li štampu ili samo na ekran?", „Hoćete li da sačuvate/snimite prethodni unos?" i slično, i „pravi" – kada se vrši popuna datoteka i sve predviđena ažuriranja.

Kod informativnih rešenja su relativno jednostavna, kao i njihovi zahtevi. Postavlja se pitanje na koje se daje kratak odgovor (najviše jedna dve reči kod identifikacije) ili D/N pitalice. Ipak, i ovde postoje sitni detalji tipa pozicioniranje (zaista nije sve jedno gde će se pitanje pojaviti i da li će ono prekriti ili istisnuti prema gore trenutno značajan detalj), ili da li se uneti odgovor vidi ili ne (kod identifikacije), a da ne spominjem da li će pitanje ili njegov deo biti naglašeni velikim slovima ili biti svetliji na ekranu. Ako se slažete sa autorom da ovo ima smisla, upotrebite saznanja iz pozicioniranja i pogledajte odeljak o mešanju boja. U svakom slučaju, ako niste sigurni, napravite i uporedite dve varijante, sa i bez ovih „sitnica". To će vam dati definitivan odgovor.

## UNOS PODATAKA

Pravi ulazni paneli su složeniji su od prethodnih jer su takvi i zahtevi pred njima. Ipak, ni to nije naročito veliki problem. To potvrđuje i nekoliko metoda za njegovo rešavanje.

Po meni, najjednostavnije je koristiti dBASE modus za izradu formata (tako ih zovu u paketu) a zatim kroz program setovati taj format i to na dva načina.

```
USE datoteka                        u itate datoteku
APPEND BLANK                        formirate novi, prazan slog
SET FORMAT TO ime formata           pozovete format
READ                                u itate unete podatke
USE                                 zatvorite datoteku
SET FORMAT TO                       isklju ite format
```

Slika 5

Prvi, naravno jednostavniji, je da napravite ulazni format za vašu datoteku/datoteke u dBASE-u i u program uvrstite prostu proceduru kao na slici 5.

U korišćenju ovog metoda, posebno vodite računa o sledećem:

- svaki novi slog mora se formirati APPEND BLANK komandom, to je bitna razlika o dBASE-a
- vaš format mora da bude na jednom ekranu, ako vam je potreban još jedan ili više, napravite i kroz program pozovite onoliko formata koliko ekrana vam treba, inače doživećete da program „podivlja"
- obavezno zatvorite datoteku i isključite format na kraju rada, a iz istih razloga kao u prethodnoj alineji
- obavezno iza svakog formata stavite READ komandu i ako ona postoji i u formatu
- poželjno je da kontrolišete poruke o greškama – korisnika treba poštovati, ali i ne očekivati previše od njega
- kad prevodite program, morate omogućiti *Clipper*-u da dođe i do FMT datoteka, inače ćete dobiti obaveštenje o grešci, a program neće funkcionisati.

Svako će se složiti da je ovo jednostavno. Ipak, postoje i određeni nedostaci. Logička kontrola unetih podataka je ograničena samo na tip polja (to ste obavili kroz formiranje datoteke), ako su u datoteku uključena i polja memo tipa pristup do njih je složen, a to ne odgovara korisniku itd.

Zbog toga postoji modalitet ovog metoda. Kreirajte format da izgleda onako kako vam

```
IF FERROR()<>0
**                                  ubacujete svoja reenje
ENDIF
```

Slika 6

```
DECLARE BOJA[10]
```

Slika 7

```
BOJA[1]="W/N,N/W,,,N/W"
BOJA[2]="W/N,N/W,,N/W"
BOJA[3]="W+/N,N/W,,N/W"
BOJA[4]="W+/N,N/W,,,N/W"
```

Slika 8

```
DECLARE EKRAN[10]  ***  ovo vam omoguuje da sa uvate 10
                        ***  razli itih ekrana istovremeno (kroz ceo
                             samo u ovom delu)
SAVE SCREEN TO EKRAN[1]
CLEAR   DO NOVI             ***  NOVI je procedura koju ubacujete (ili
                            ***  HELP meni)
CLEAR
RESTORE SCREEN FROM EKRAN[1]
```

Slika 9

odgovara. Zatim, pozovite, iz editora koji koristite, FMT datoteku i prema njoj se odnosite kao da je deo vašeg programa. Između GET-ova ubacujte svoju logičku kontrolu ili bilo koje druge petlje koje nađete za shodno. Potom, kopirajte (merdžujte) FMT datoteku kao integralni deo programa i time ste završili dobar posao. Posle toga, slobodno možete da izbrišete FMT i SCR datoteke iz direktorijuma.

Sigurno ste uočili da bi ovo mogao da bude još jedan način za izradu i crtanje menija (naravno bez GET-ova).

Sve komande za prihvat podataka i ažuriranje datoteka, možete, naravno, i sami da smišljate i pišete, koristeći pozicioniranje SAY/GET i ostala iskustva i znanja. Nekad, ali zaista vrlo retko, ovo je jedino rešenje.

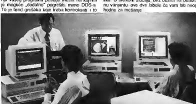
## UPOZORENJE NA GREŠKE

Ovaj segment iznenađujuće mnogo programera prepušta DOS-u. Ne bi bilo ni primereno ni neophodno ulaziti u raspravu koliko je to opravdano i dobro. Elem, DOS ima svoj fond grešaka koje detektuje i prijavljuje. U korišćenju vašeg programa postoje mesta na kojima je moguće „dodatno" pogrešiti, mimo DOS-a. To je fond grešaka koje treba kontrolisati i to baš po toj sistematizaciji: DOS ili PRG. Kad se radi o vašem programu, moraćete sami da se potrudite i oko rešenje i oko predviđanja. Što se DOS-a tiče, velika pomoć je tabela DOS grešaka, odnosno tabela numeričkih vrednosti koja prate DOS poruke o grešci. Da bi ova tabela mogla i da vam koristi, potrebno je da instalirate i proceduru sa slike 6(znajući, pri tom, da FERROR() vraća 0 ako greška ne postoji).

U svoje rešenja možete da uvrstite prevod poruke DOS greške, petlju za povratak na mesto gde je greška napravljena i slično.

## KORIŠĆENJE BOJA

Kao i u svakom drugom poslu, mešanje boja po ekranu će vam biti mnogo lakše ako prethodno dobro predvidite i postavite potrebe koje ćete imati tokom programa. Kako se to ovde radi? Podešavanje boja na ekranu izvodi se u *Clipper* u pomoću dve komande, SET COLOR TO i SETCOLOR(niz). Razlika je u tome što sa drugom koristite već deklarisani niz karaktera (koji predstavlja jednu od kolor varijanti), dok iza prve uvek morate da navedete taj niz (ili da koristite &), što i ne predstavlja bitnu razliku. U svakom slučaju, bez obzira na odabranu varijantu, ove dve labele će vam biti neophodne za mešanje.



| Za kolor karticu/monitor: | |
|---|---|
| BOJA | SLOVO |
| crna | N |
| bela | W |
| plava | B |
| zelena | G |
| braon | GR |
| ljubičasta | RB |
| žuta | GR+ |
| crvena | R |
| cijan | BG |
| siva | N |

| Za monohromatsku varijantu: | |
|---|---|
| BOJA | SLOVO |
| crna | N |
| bela | W |
| podvučeno | U |
| inverzno | I |

Da biste istakli/pojačali neku boju, dodajete joj znak plus (+) koji će pojačati osvetljaj. Da bi deo teksta na ekranu, ili ceo ekran (što nije baš najprijatnije za onog ko gleda) blinkovali/treptali, slovu dodajte zvezdicu (*).

Kako će to u programu da izgleda? U varijanti SETCOLOR(), neophodno je deklarisati varijable koje će biti korišćene kao nazivi pojedinih kolor varijanti. Najkraće je kao na slici 7.

čime smo dobili deset naziva na raspolaganje. Zatim treba opisati kombinacije. Nekoliko primera je dato na slici 8.

Nakon toga, sve je prepušteno vašoj mašti, a savet je da pažnju obratite na sledeće:

- nije neophodno, a suprotno je i preporučljivo, brisati ekran kod menjanja seta boja. Time se najlakše i ističe deo ekrana/tekst zbog koga se set i menja,
- SET COLOR TO bez daljeg navođenja vraća osnovni/početni set boja i treba ga koristiti nakon postizanja željenog efekta sa posebnim varijantama,
- nemojte biti neumereni u mešanju boja i šaranju ekrana, jer to može da izazove i suprotan efekat i da vam ekran učini nepreglednim i nerazumljivim.

## ČUVANJE EKRANA

Ovo je neophodna i relativno jednostavna operacija unutar programa. Pomislite samo koliko problema (izraženo u pisanju procedura i gomile programskih redova) bi stvarao tzv. HELP meni prilikom restauracije ekrana (za one koji nisu sigurni, to je uputstvo za korišćenje programa i pružanje pomoći u njegovom korišćenju unutar samog programa, obično se poziva pritiskom na F1-funkcijski taster i preko teksta ispisanog na ekranu dostavlja informacije o programu). Ovako, jednostavno:

– sačuvate izgled ekrana,
– ispišete sadržaj koji želite,
– pozovete sačuvani ekran.

Pretpostavljamo da ćete pokatkad osetiti potrebu i da obrišete ekran. To će obaviti procedura na slici 9.

Ceo ovaj tekst i ponuđeni primeri imaju za cilj da vam pomognu oko rešavanja pojedinih „strateških" pitanja postavke vaših programa, a do finala vam predstoji još dosta dug put.

# TAMANITE VIRUSE A NE TETRISE

**Zašto su virusi danas brojniji, aktivniji i moćniji nego ikada? Zašto veliki računarski sistemi sve više padaju pod naletima sve „pametnijih" virusa? Iz dva razloga: zbog nestašne mašte jednih i nezainteresovanosti drugih programera. Kada zec spava, kornjača dobija trku.**

Jovan Sedlan

Probudite, dakle, zeca. Stavite mu šargarepu pred nos i naterajte ga da koristi ono čime ga je priroda obdarila. Ovim tekstom želeo bih da apelujem na sve zdrave mozgove koji ovo čitaju: ostavite *Tetris* i ubijajte viruse! Moj doprinos ovoj paroli zove se

## DOT BOUNCER

Verovatno ste već čuli o računarskim *boot* virusima. To su virusi koji se šire preko *boot* sektora diskete ili hard diska. Jedan od najpoznatijih kod nas je *ping pong*, alijas *italian*, alijas *bouncing dot*, alijas *loptica*.

Da malo razjasnimo ovu gomilu imena: prva dva i poslednji naziv su sasvim jasni i za one koji nisu upoznati sa lepotama engleskog jezika. *bouncing dot* bi, u slobodnom prevodu glasilo *loptica skočica*, s tim da *dot* uopšte nije „loptica", već „tačka" (pogodili ste – upravo zato se slobodan prevod zove tako kako se zove). Sve ovo, naravno, nimalo ne razjašnjava gornji podnaslov, ali ako kažemo da *bouncer* (izbacivač u kafani) i *bouncing* (skakutanje) u suštini nemaju mnogo veze, onda slika postaje jasnija: *dot bouncer* je „izbacivač loptice". Naravno, radi se o programu koji leči vaš računar neželjene bolesti Đ

Ako ste kojim slučajem „običan" korisnik računara koma „tamo neka bedna loptica dosađuje s vremena na vreme (i to uvek u nezgodnom trenutku)", ne morate se ni truditi da čitate sledeći tekst, sve što treba da uradite je da program koji je ovde objavljen otkucate u nekom editoru i zatim ga asemblirate sa Đ

```
MASM BOUNCER,
LINK BOUNCER
EXE2BIN BOUNCER EXE BOUNCER COM
```

Ukoliko umesto MASM-a koristite *Turbo Assembler*, onda otkucajte

```
TASM BOUNCER
TLINK /t/x BOUNCER
```

Dot Bouncer se startuje sa

```
BOUNCER d
```

pri čemu umesto d treba da stavite ime diska koji proveravate, odnosno lečite.

Za sve one kojima programerski duh ne da mira i stalno traži nove izazove (znači za one koji nisu „obični" korisnici računara) sledi ono (zanimljiviji deo, u stvari) što sam uspeo da otkrijem o ovom izuzetno interesantnom i lukavom virusu. Osim toga, ovaj tekst može mnogo da pomogne u rešavanju problema koje imate ili možete imati sa nekim drugim *boot* virusom (nijedan drugi *boot* virus ne može se mnogo udaljiti od filozofije „loptice" koja je tipičan predstavnik svoje vrste).

## DA KRENEMO

Radi se o virusu koji inficira PC XT/AT i kompatibilne računare (odnosno diskove koji se na njima koriste). O ovom virusu je već pisano u domaćim računarskim časopisima, ali još nigde nije objavljen lek (izgleda da je PC i u pogledu ove vrste softvera shvaćen kao računar koji služi čisto za zaradu).

Ako vam ne bude suviše teško da unesete program koji je ovde objavljen i da ga zatim asemblirate, *verovatno* ćete se zauvek osloboditi ovog virusa. Ono „verovatno" nalazi se u prethodnoj rečenici upravo zato što, kako vam je već sigurno poznato, „zdravlje" vašeg računara ne zavisi samo od *anti virus* programa koje posedujete, nego i od vaše dobre volje i potrebe da svoj računar na najbolji način zaštitite. Upravo takvoj vrsti zaštite je potrebno posvetiti najviše pažnje i zato krećemo od samog početka.

## ZDRAV DISK...

Da biste shvatili način na koji ovaj virus radi, neophodno je najpre upoznati se sa DOS organizacijom diska. O ovome je već pisano u ranijim brojevima „Računara", ali ćemo ovde sve to ukratko ponoviti. Komunikacija programera i diska se obavlja uz pomoć DOS-a, ili na nižem nivou, preko BIOS-a. Naravno, za programera je ona prva dosta lakša za korišćenje, ali zato ima i određen broj nedostataka u odnosu na drugu. Što se BIOS-a tiče, svaki disk (disketa/hard disk) ima X strana (glava), Y traka i Z sektora po traci. Na osnovu ovih podataka BIOS čita ili piše odgovarajući sektor na disk. Na nivou DOS-a, međutim, uvodi se pojam logičkih sektora. Broj ovih sektora zavisi od kapaciteta diska. Prvi logički sektor ima broj 0, drugi je broj 1 itd., do kraja diska. Sve dovde nema neke značajne razlike između ova dva pristupa čitanju i pisanju na disk. Prva značajnija razlika se javlja upravo u različitom tretiranju hard diskova.

BIOS tretira hard disk baš kao i najobičniju disketu: ima njegov broj i na osnovu tri koordinate (cilindar/glava/sektor) pristupa određenom sektoru. Sa druge strane, DOS omogućava podelu velikih hard diskova na više manjih. Za tako nešto potrebno je na disku zauzeti mesto za labelu koja će čuvati podatke o svim manjim diskovima koji se u terminologiji DOS-a nazivaju particije (zbog toga se ova tabela naziva labela particija). To mesto se nalazi upravo na samom početku diska (cilindar 0, glava 0, sektor 1). Inače, ova labela postoji na svakom hard disku i zauzima celu nultu stazu na strani 0, bez obzira na činjenicu da je za nju dovoljan i samo jedan sektor.

Zahvaljujući labeli particija, DOS pristupa (na primer) hard disku D baš kao i svakom drugom disku, ako međutim pokušamo ovom disku da pristupimo preko BIOS-a, ovaj će nam javiti da nije uspeo da pročita (ili upiše) baš ništa. Naravno, sve to pod uslovom da je hard disk izdeljen na particije i da je disk D jedna od njih. Interesantno je i to da DOS ne dozvoljava pristup labeli particija (što je i prilično razumljivo) – logički sektor sa brojem 0 se nalazi na strani 1, cilindar 0, sektor 1. Prema tome, slika hard diska izgleda ovako:

- Tabela particija
- *boot* sektor
- FAT tabele (najčešće dve)
- ROOT direktorijum
- Prostor za podatke

Naravno, ovde se radi o hard disku koji **nije** izdeljen na particije. U slučaju hard diska koji ima X particija, sve ovo (osim tabele particija) se ponavlja još X – 1 puta.

## ...I DIZANJE SISTEMA

Kako se „diže sistem" na PC-u? Najpre se učita tabela particija (ako se radi o disketi, odmah se učitava *boot* sektor i predaje mu kontrola). Na njenom početku se nalazi program koji u labeli pronalazi koja od particija je „bootable" (particija sa koje može da se „podigne sistem"), učitava *boot* sektor te particije i počne da izvršava program koji je u njega upisan. Dalje sve zavisi od operativnog sistema. Ako je u pitanju DOS, onda se u *boot* sektoru nalazi program koji radi sledeće:

1. Proverava da li se IBMBIO.COM i IBMDOS.COM nalaze u *root* direktorijumu. Ova dva fajla moraju biti prva dva fajla u direktorijumu i moraju se nalaziti upravo u tom redosledu (ovo je za slučaj PC DOS-a; ako se radi o MS DOS-u, onda se ova dva fajla zovu IO.SYS i MSDOS.SYS).
2. Učitava IBMBIO.COM (IO.SYS) u memoriju i predaje mu kontrolu.
3. Učitava se IBMDOS.COM, utvrđuje status sistema, resetuje se disk sistem, inicijalizuju se priključeni uređaji, učitavaju se drajveri, postavljaju se vrednosti interapt vektora, relocira se IBMDOS.COM naniže u memoriju i predaje mu se kontrola.
4. DOS inicijalizuje interne radne tabele, inicijalizuje interapt vektore za interapte od 20h do 27h i kreira PSP (Program Segment Prefix) za COMMAND.COM na najnižem slobodnom segmentu. U DOS verzijama od 3.10 do 3.30 DOS inicijalizuje interapt vektore za interapte od 0Fh do 3Fh.
5. Poziva se COMMAND.COM uz pomoć EXEC funkcije DOS-a i predaje mu se kontrola.

Naravno, sve se ovo dešava u savršeno normalnim okolnostima, tj. kada je disk „zdrav". Međutim, priličan deo priče je drugi kada sve to umeša svoje prste „punokrvni" virus kakav je „loptica". Što se diska tiče, pitanje na koje još uvek nismo odgovorili glasi:

## KAKO GA ZARAZITI?

Objasnićemo način na koji se ovaj virus širi. Ako, na primer, imamo računar sa hard diskom koji nije inficiran ovim virusom, tada ga

## Listing 3

```
TITLE  Dot Bouncer

wp              EQU     word ptr
bell            EQU     7
CR              EQU     13
LF              EQU     10
new_line        EQU     CR, LF
str_end         EQU     '$'
retry           EQU     5

boot_info       STRUC
                jump                    db      3 dup (0)
                o_e_m                   db      8 dup (0)
                bytes_per_sector        dw      0
                sectors_per_cluster     db      0
                reserved_sectors        dw      0
                number_of_FATs          db      0
                number_of_root_entries  dw      0
                total_sectors           dw      0
                media_descriptor        db      0
                FAT_sectors             dw      0
                sectors_per_track       dw      0
                number_of_heads         dw      0
                number_of_hidden_sect   dw      0
boot_info       ENDS


CODE            SEGMENT
                ASSUME  CS: CODE, DS: CODE
                ORG     100h
begin           jmp     main


orig_sector     dw      0
drive           db      0
FAT_sector      dw      0
FAT_offset      dw      0
cluster_number  dw      0
bit_mask        dw      0
sector_number   dw      0
buffer          db      512 dup (0)
info            boot_info
boot_pattern    db      33h, 0C0h, 8Eh, 0D0h, 0BCh, 00h, 7Ch
                db      8Eh, 0D8h, 0A1h, 13h, 04h, 2Dh, 02h
                db      0Dh, 0A3h, 13h, 04h, 0B1h, 06h, 0D3h
                db      0E0h, 2Dh, 0C0h, 07h, 8Eh, 0C0h, 0BEh
                db      00h, 7Ch, 8Bh, 0FEh, 0B9h, 00h, 01h
                db      0F3h, 0A5h, 06h, 0C0h, 0Eh

system_pattern  db      1Eh, 06h, 50h, 53h, 51h, 52h, 56h
                db      1Fh, 0E8h, 07h, 0F6h, 0Eh, 0F7h, 7Dh
                db      01h, 75h, 43h, 0Dh, 0FCh, 03h, 75h
                db      3Dh, 2Eh, 18h, 0F8h, 70h, 0Bh, 16h
                db      0FBh, 7Dh, 75h, 22h, 32h, 0E4h, 0CDh
                db      1Ah, 0F6h, 0C7h, 7Fh, 75h

program_name    db      'Dot Bouncer  Version 1.0', new_line
                db      'Ping Pong/Bouncing Dot/Italian  VIRUS KILLER  1990'
                db      new_line, LF, str_end
usage           db      'You must specify the drive name (A:, B:, C:, etc.)'
                db      bell, new_line, str_end
DOS_version     db      'Wrong DOS version!', bell, new_line, str_end
infected_disk   db      'Disk A: is infected!', bell, new_line, str_end
read_error      db      'Read error!', bell, new_line, str_end
write_error     db      'Write error!', bell, new_line, str_end
boot_recovered  db      'BOOT successfully recovered!', new_line, str_end
FAT_recovered   db      'FAT successfully recovered!', new_line, str_end
sound_disk      db      'Disk A: is not infected!', new_line, str_end
restart         db      'System will restart now!', new_line, LF
                db      'Press any key to continue...', str_end

main            PROC    NEAR
                xor     ax, ax
                push    ax                      ; Uskladi stek
                call    initialize              ; Inicijalizacija
                call    get_parameters          ; Proveri komandnu liniju
                call    check_boot              ; Da li je disk zaražen?
                call    recover_FAT             ; Da objasnimo ovaj
                                                ;  redosled: da smo
                                                ;  najpre popravili BOOT
                                                ;  sektor pa tek onda
                                                ;  FAT tabele, došlo bi
                                                ;  do ponovnog čitanja
                                                ;  što bi prouzrokovalo
                                                ;  da se disk ponovo zarazi
                call    recover_boot            ; Opravljanje BOOT sektora
                call    check_system            ; Ukoliko se virus nalazi
                                                ;  u memoriji, obavezan je
                                                ;  reset sistema
the_end:        pop     ax
                or      ax, ax
                jne     the_end                 ; Očisti stek
                mov     ax, 4C00h
                int     21h                     ; Završi sa radom
main            ENDP

initialize      PROC    NEAR
                lea     dx, program_name
                call    print_string            ; Ispiši ime programa
                mov     ah, 30h
                int     21h                     ; Proveri DOS verziju
                or      al, al                  ; Da li je verzija 1.0?
                jnz     maybe_ok                ;  ako nije, skoči
wrong_version:  lea     dx, DOS_version         ; Pogrešna verzija,
                jmp     fatal_error             ;  završi program
maybe_ok:       cmp     al, 3                   ; Da li je verzija 4.XX?
                je      wrong_version           ;  ako jeste, pogrešna je
ok:             ret                             ;  nije, sve je OK
initialize      ENDP


get_parameters  PROC    NEAR
                cld
                mov     si, 0080h
                lodsb                           ; Proveri da li je komandna
                or      al, al                  ;  linija prazna
                jnz     go_on                   ;  ako jeste, nastavi
how_to_use:     lea     dx, usage               ;  inače, ispiši poruku
                jmp     fatal_error             ;  greške i završi
go_on:          lodsb
                cmp     al, ' '
                je      go_on
                mov     ah, al                  ; Oznaka drajva u AH
                lodsb                           ; Da budemo sigurni da se
                cmp     al, ':'                 ;  radi o disk jedinici
                jnz     how_to_use
                and     ah, 11011111b           ; Pretvori u veliko slovo
                mov     infected_disk+5, ah
                mov     sound_disk+5, ah
                and     ah, 10111111b
                dec     ah                      ; Drajv koji se proverava,
                mov     drive, ah               ;  odnosno leči
                ret
get_parameters  ENDP

check_boot      PROC    NEAR
                xor     dx, dx                  ; DX = 0
                call    read_sector             ; Čitaj nulti sektor
                xor     bx, bx                  ; Ako se na početku BOOT
                mov     ax, wp buffer[bx]       ;  sektora ne nalazi
                cmp     ax, 1CEBh               ;  1CEBh, onda disk
                jz      continue                ;  sigurno nije zaražen
not_infected:   lea     dx, sound_disk          ; Disk je OK
                jmp     fatal_error
continue:       cld                             ; Proveri da li je disk
                mov     cx, 20                  ;  zaražen na osnovu 40
                lea     di, buffer + 7Ch        ;  bajtova karakterističnih
                lea     si, boot_pattern        ;  za ovaj virus
                repe    cmpsw
                jne     not_infected            ; Disk je zdrav
                lea     si, buffer              ; Disk je zaražen, prebaci
                lea     di, info                ;  30 bajtova iz bafera u
                mov     cx, 15                  ;  prostor za podatke
                rep     movsw                   ;  (ovih 30 bajtova sadrže
                mov     bx, 1F9h                ;  podatke o disku koji će
                mov     dx, wp buffer[bx]       ;  nam kasnije biti
                inc     dx                      ;  neophodni)
                mov     orig_sector, dx         ; Mesto na disku gde je
                                                ;  smešten originalni
                                                ;  BOOT sektor
                lea     dx, infected_disk
                call    print_string            ; Ispiši poruku
                ret
check_boot      ENDP

recover_FAT     PROC    NEAR
                mov     ax, info.FAT_sectors
                mov     bl, info.number_of_FATs
                mul     bl
                mov     bx, ax
                mov     ax, info.number_of_root_entries
                mov     cl, 4
                shr     ax, cl
                mov     cx, ax
                mov     dx, info.reserved_sectors
                mov     ax, orig_sector
                sub     ax, dx
                sub     ax, cx
                sub     ax, bx
                xor     bx, bx
                mov     bl, info.sectors_per_cluster
                xor     dx, dx
                div     bx
                add     ax, 2                   ; U AX registru se sada
                                                ;  nalazi broj klastera
                                                ;  u kome se nalazi
                                                ;  originalni BOOT sektor
                push    ax                      ; Sačuvaj AX
                mov     ax, info.total_sectors
                add     ax, info.number_of_hidden_sect
                cmp     ax, 20740               ; Proveri da li je FAT
                pop     ax                      ;  tabela 12-bitna ili
                ja      bit_16                  ;  16-bitna
```

Listing se nastavlja na sledećoj strani

## Nastavak sa prethodne strane

```
bit_12:         mov     bx, ax
                clc
                rcr     ax, 1
                jc      hi
lo:             mov     bit_mask, 0F000h
                add     ax, bx
                jmp     r_f_1
hi:             mov     bit_mask, 000Fh         ; Izracunaj BIT MASKU za
                add     ax, bx                  ;  za 12-bitni FAT
                jmp     r_f_1
bit_16:         shl     ax, 1
                mov     bit_mask, 0             ; BIT MASKA za 16-bitni FAT
r_f_1:          call    FAT_sect_ofs            ; Pronadji sektor i ofset
                                                ;  u FAT tabeli
                mov     ax, bit_mask
                and     wp buffer[bx], ax       ; Oslobodi klaster
                mov     dx, FAT_sector
                call    write_sector            ; Upisi prepravljeni FAT
                xor     cx, cx
                mov     cl, info.number_of_FATs
                dec     cl
                jcxz    no_more_FATs
r_f_2:          push    cx
                mov     dx, FAT_sector
                add     dx, info.FAT_sectors
                mov     FAT_sector, dx
                call    read_sector
                mov     dx, FAT_sector
                mov     bx, FAT_offset
                mov     ax, bit_mask
                and     wp buffer[bx], ax
                call    write_sector
                pop     cx
                loop    r_f_2                   ; Prepravi sve kopije FAT
no_more_FATs:   lea     dx, FAT_recovered
                call    print_string            ; Ispisi poruku
                ret
recover_FAT     ENDP

FAT_sect_ofs    PROC    NEAR
                mov     dx, ax
                and     ax, 111111111b          ; Ofset u AX
                mov     FAT_offset, ax
                mov     cl, 9
                shr     dx, cl                  ; Sektor u DX
                add     dx, info.reserved_sectors
                mov     FAT_sector, dx          ; Broj logickog sektora
                call    read_sector             ; Cita odgovarajuci sektor
                mov     bx, FAT_offset          ; U BX se smesta ofset
                                                ;  reci koja oznacava
                                                ;  los klaster
                ret
FAT_sect_ofs    ENDP

recover_boot    PROC    NEAR
                mov     dx, orig_sector
                call    read_sector             ; Ucitaj originalni BOOT
                xor     dx, dx
                call    write_sector            ; Upisi ga na pocetak diska
                lea     dx, boot_recovered
                call    print_string            ; Ispisi poruku
                ret
recover_boot    ENDP

check_system    PROC    NEAR
                xor     ax, ax                  ; Ova procedura radi veoma
                mov     ds, ax                  ;  jednostavan posao:
                mov     ax, ds:[0413h]          ; proverava da li je
                mov     cl, 6                   ; virus u memoriji
                shl     ax, cl                  ; racunara i, ukoliko
                mov     es, ax                  ; jeste, sistem se
                mov     di, 0000h               ; resetuje
                lea     si, system_pattern
                cld
                mov     cx, 20
                repe    cmpsw
                jnz     ok_to_return            ; Sistem nije zarazen
                lea     dx, restart
                call    print_string            ; Ispisi poruku
                xor     ax, ax                  ;  Sacekaj pritisak
                int     16h                     ;  na bilo koji taster
                mov     ax, 0FFFFh
                push    ax
                inc     ax
                push    ax
                retf                            ; Resetuj sistem
ok_to_return:   ret
check_system    ENDP

read_sector     PROC    NEAR
                mov     sector_number, dx       ; Broj logickog sektora
                mov     cx, retry               ; U slucaju greske,
read_retry:     push    cx                      ;  pokusaj ponovo CX puta
                mov     dx, sector_number
                mov     al, drive
                mov     cx, 1                   ; Cita se jedan sektor
                lea     bx, buffer              ; Adresa bafera u BX
                int     25h                     ; DOS poziv za citanje
                pop     dx                      ; Skini FLAG rec sa steka
                pop     cx
                jnc     read_ok
                loop    read_retry
                lea     dx, read_error          ; Greska u citanju
                jmp     fatal_error
read_ok:        ret                             ; Uspesno procitan sektor
read_sector     ENDP
write_sector    PROC    NEAR
                mov     sector_number, dx       ; Procedura veoma slicna
                mov     cx, retry               ;  proceduri read_sector
write_retry:    push    cx                      ;  s tom razlikom da se
                mov     dx, sector_number       ; ovde radi o pisanju
                mov     al, drive               ; jednog sektora
                mov     cx, 1
                lea     bx, buffer
                int     26h
                pop     dx
                pop     cx
                jnc     write_ok
                loop    write_retry
                lea     dx, write_error
                jmp     fatal_error
write_ok:       ret
write_sector    ENDP
fatal_error     PROC    NEAR
                call    print_string            ; Ispisi poruku i
                jmp     the_end                 ;  zavrsi program
fatal_error     ENDP
print_string    PROC    NEAR
                mov     ah, 9                   ; Pozovi DOS funkciju 9
                int     21h                     ;  za ispis stringa
                ret
print_string    ENDP
code            ENDS
                END     begin
```

**Program za uklanjanje virusa „Loptica skočica"**

možemo zaraziti na sledeći način: potrebna nam je, pre svega, inficirana disketa. Ta disketa može biti sistemska ili ne. U prvom slučaju, kada dignemo sistem sa nje, doći će do inficiranja računara i pri svakom sledećem čitanju bilo kog diska dolazi do njegove infekcije. Ovde treba upozoriti na to da čak i najobičnije zapisivanje u datoteku uzrokuje prethodno čitanje FAT tabele, direktorijuma itd., pri čemu takođe dolazi do infekcije.

U slučaju da dotična disketa nije sistemska, pri pokušaju da se digne sistem, najpre će se inficirati računar i zatim ispisati poruka tipa „Non System disk or disk error. Replace and strike any key when ready". Ako učinimo to što nam poruka kaže, doći će do infekcije najpre diska kojim smo zamenili ne-sistemsku disketu (ili hard diska ako smo jednostavno izvadili disketu i pritisnuli taster), a zatim i svakog drugog diska sa kojeg bude čitano.

## INFICIRAN DISK...

Kada smo tako uspešno zarazili disk (neuspesi su isključeni – pitajte Marija), da vidimo kako sve to zajedno izgleda. Kao što smo rekli, ovo je virus *boot* tipa, što znači da virusni kod snima u *boot* sektor. Ono što karakteriše svaki virus ovog tipa je činjenica da je potrebno negde na disku sačuvati originalni sadržaj *boot* sektora. Pitanje koje pri tom sebi postavlja autor *boot* virusa je: kako pronaći slobodno mesto na disku i na njega snimiti *boot* sektor i kako obezbediti da nijedan drugi program ne upiše novi sadržaj na isto mesto?

U slučaju „loptice", autor ovog virusa je osmislio sledeće rešenje (za koje se baš i ne može reći da je originalno): uz pomoć FAT tabele virus pronalazi prvi slobodni klaster na disku, označi ga kao loš (bad cluster – FF7h za 12-bitni FAT, odnosno FFF7h za 16-bitni) da DOS ne bi kojim slučajem „brljao" po njemu i zatim u njega snimi najpre jedan sektor virusnog koda, a zatim i originalni *boot* sektor. Sve u svemu, jedan klaster je dovoljan za snimanje ova dva sektora (na hard diskovima jedan klaster najčešće sadrži četiri sektora, a na disketama dva).

NASTAVLJA SE NA STRANI 82

# ARITMETIČKI IZRAZI

**Čak i kada imamo napisane sve potprograme za obavljanje operacija i izračunavanje funkcija, rešavanje zadataka u mašinskom jeziku daleko je od udobnog. Teško da bilo šta može zameniti komfor jedne formule zapisane u nekom od viših programskih jezika. U okviru našeg programa za računanje trenutno takav komfor nemamo, ali nam ništa ne stoji na putu da ga ostvarimo. Napisaćemo program za interpretaciju aritmetičkih izraza.**

Jovan Skuljan

Ma koji od viših programskih jezika da koristimo, veoma jednostavno ostvarujemo računanje bilo kakve formule: jer tu formulu prosto ukucavamo u program gotovo isto onako kako je pišemo na papiru. Da li smo se ikada upitali kako naš kompajler (ili interpreter) izlazi na kraj sa takvim formulama kada znamo koliko tu ima komplikacija (zagrade, prioriteti binarne operacije, unarne operacije i tako dalje)? A da li smo se ikada odvažili da sami napišemo program koji bi obavljao takav posao kao što je vrednovanje aritmetičkih izraza? Ako nismo, sada je vreme da to uradimo.

Pre svega, moramo jasno formulisati svoje želje. Hoćemo da negde u memoriji, na poznatoj adresi, upišemo formulu u obliku ASCII znakova, slovo po slovo, kao u bejziku, fortranu ili paskalu. Voleli bismo da u toj formuli figurišu brojevi, funkcije, zagrade i promenljive. Odvojićemo poseban blok memorije za promenljive gde će svaka imati svoje ime i vrednost. Na primer, promenljiva VREME može imati vrednost 7.3, pa bi rezultat izraza VREME/2 bio 3.65. Dopustićemo, takođe, promenljive tipa niza sa jednom, dve ili više dimenzija. Ne znamo još kako ćemo sve te promenljive organizovati, ali računamo na njihovu upotrebu.

Pošto je zadatak jasen, pristupamo njegovom rešavanju. Počećemo detaljnom analizom sastavnih delova aritmetičkog izraza, a onda preći na razvoj algoritma za izračunavanje.

## OPERACIJE I OPERANDI

U najjednostavnijem obliku, aritmetički izraz je niz *operanada* međusobno razdvojenih *operacijama*, kao na primer *A+B*C/D* pri čemu svaki operand (svako slovo u navedenom primeru) označava neku *fiksnu brojnu vrednost*. Ta brojna vrednost može biti zadata u vidu obične numeričke *konstante* (recimo 2.71) u vidu imena neke *promenljive* (na primer VREME) ili u vidu neke elementarne matematičke *funkcije* (SIN, COS, TAN, EXP i tako dalje).

U nešto složenijoj formi, na mestu operanda može stajati *otvorena zagrada*, što je znak da sledi aritmetički izraz čiju vrednost treba računati zasebno, kao jednu celinu. Na kraju takvog „podizraza" obavezno stoji *zatvorena zagrada*.

Međutim, u bilo kom slučaju, jasno je da aritmetički izraz mora početi i mora se završiti *operandom*, bez obzira na to u kom obliku se operand pojavljuje.

## BINARNE OPERACIJE

Posle ove grube skice, možemo preći na detaljniju analizu pojedinih elemenata aritmetičkog izraza. Pre svega ćemo razmotriti operacije.

Iz onoga što smo do sada rekli, jasno je da imamo u vidu isključivo *binarne operacije*, jer se javljaju uvek između dva operanda. *Unarne operacije*, kakva je, recimo, negativan predznak (deluje samo na operand iza sebe), tretiraćemo prosto kao funkcije (što one i jesu), a to znači kao *operande* (na način koji smo već objasnili).

Binarnih operacija nema mnogo. Tokom ove serije bavili smo se detaljno sa četiri osnovne aritmetičke operacije: + (sabiranje), − (oduzimanje), * (množenje) i / (deljenje). Ovom spisku ćemo još dodati i operaciju (stepenovanje), mada još nemamo odgovarajući potprogram u okviru kalkulatora (uskoro ćemo se i time pozabaviti). Isto tako, uvešćemo i korisnu, mada ne baš naročito standardnu, operaciju stepenovanja *celobrojnim izložiocem*, koja se izvršava jednostavno uzastopnim množenjem osnove potreban broj puta. Na primer, $2^3$ je 2*2*2.

Jasno, mogli bismo $2^3$ izračunati koristeći operaciju 2 3, ali bi to bilo nepotrebno gubljenje vremena. Operacija stepenovanja, kao što ćemo to videti kada budemo za nju pisali program, ne izvršava se baš naročito jednostavno, jer se podrazumeva da izložilac može biti proizvoljan realan broj. A dizanje na kvadrat ili na treći stepen su ipak česte i proste radnje, tako da se isplati uvesti za njih posebnu operaciju.

Stepenovanje celobrojnim izložiocem označićemo znakom ⋕ i definisaćemo ga kao $a⋕n=a*a*a...*a$, gde se osnova a množi sama sobom ukupno *n* puta. Podrazumeva se da je *n* prirodan broj.

Konačan spisak binarnih operacija koje se mogu javiti u okviru aritmetičkog izraza bio bi:

| | | |
|---|---|---|
| + | *sabiranje* | (ADDITION) |
| − | *oduzimanje* | (SUBTRACT) |
| * | *množenje* | (MULTIPLY) |
| / | *deljenje* | (DIVISION) |
| | *stepenovanje* | (POWER) |
| ⋕ | *celobrojno stepenovanje* | (INTPOWER) |

U zagradama su dati potprogrami koji obavljaju zadatu operaciju. Potprograme POWER i INTPOWER daćemo u sledećem nastavku serije. Za sada nam je jedino važno da svi pobrojani potprogrami podrazumevaju da se oba operanda nalaze na aritmetičkom steku, pri čemu je drugi operand (umanjilac, delilac ili izložilac) u samom vrhu steka, dok je prvi operand ispod njega.

Na operacije ćemo se vratiti i kasnije, kada budemo analizirali konkretan algoritam vrednovanja aritmetičkog izraza, a sada možemo da se posvetimo glavnim sadržinskim elementima izraza: operandima.

## OPERANDI I FUNKCIJE

Kao što smo već rekli, aritmetički izraz mora početi operandom. Na taj način, pitanje *šta sve može biti operand* ekvivalentno je sa pitanjem *čime sve može početi aritmetički izraz*. Najopštije gledano, izraz može započeti isključivo na jedan od sledeća tri načina:

1. Numeričkom vrednošću
2. Unarnom operacijom (funkcijom)
3. Otvorenom zagradom

Pod *numeričkom vrednošću* može se javiti ili neki *broj* (kao u izrazu *2*R*PI*) ili neka *promenljiva* (kao u izrazu *R⋕2*PI*).

Pod *unarnom operacijom* dopuštamo samo sledeće operacije (odnosno funkcije):

| | | |
|---|---|---|
| 1 | + | *Predznak plus* |
| 2 | − | *Predznak minus* |
| 3 | **ABS** | *Apsolutna vrednost* |
| 4 | **INT** | *Celobrojna vrednost* |
| 5 | **SQRT** | *Kvadratni koren* |
| 6 | **SIN** | *Sinus* |
| 7 | **COS** | *Kosinus* |
| 8 | **TAN** | *Tangens* |
| 9 | **ASIN** | *Arkus sinus* |
| 10 | **ACOS** | *Arkus kosinus* |
| 11 | **ATAN** | *Arkus tangens* |
| 12 | **LN** | *Prirodni logaritam* |
| 13 | **EXP** | *Eksponencijalna funkcija* |
| 14 | **SGN** | *Funkcija znaka* |

Iza unarne operacije, situacija je ista kao i na početku aritmetičkog izraza. Recimo, iza predznaka minus može doći numerička vrednost (−5 ili −X), ili nova unarna operacija, kao −sin(1), ili otvorena zagrada. To što važi za predznak minus, važi i za bilo koju operaciju iz gornjeg spiska. Međutim, iz praktičnih razloga, iza imena funkcije ima smisla staviti samo otvorenu zagradu, pošto kombinacije tipa SIN1 ili SINX izazivaju nedoumicu da li je to SIN(1) i SIN(X) ili se radi o imenima nekih varijabli? Kada budemo detaljno opisivali algoritam za izračunavanje izraza, opredelićemo se za takvo rešenje da iza imena funkcije ne može stajati nikakav alfanumerički karakter. Ali o tome će tek biti diskusije.

I konačno, izraz može početi zagradom. U tom slučaju, iza zagrade važe pravila kao i na početku izraza, s tim što se čitav podizraz mora završiti zatvorenom zagradom, što onda predstavlja jedan jedinstven operand. Na primer, možemo smatrati da izraz (2+3)*4 ima dve operanda, od kojih je prvi (2+3), a drugi 4.

## SA GLEDIŠTA MIKROPROCESORA

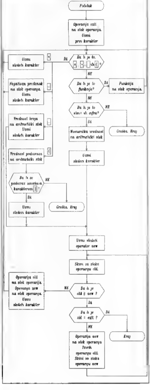
Definisanjem početka izraza mi smo praktično definisali ceo izraz, jer iza prvog operanda dolazi neka operacija, pa novi operand i tako dalje. Svaki operand, dakle, može biti:

1. Numerička vrednost
2. Funkcija
3. Podizraz

Ako iza operanda ne stoji operacija, to je oznaka kraja aritmetičkog izraza.

Sve je to lepo, ali da smo mi na mestu jednog mikroprocesora, i dalje ne bismo umeli da izračunamo ni jedan jedini aritmetički izraz. Šta nam mogu značiti pojmovi *numerička konstanta*, *funkcija* ili *podizraz*, kada pri čitanju izraza

## Slika 1




## Listing 1 Izračunavanje aritmetičkog izraza

Konstante i promenljive

```
VARS       DB    2048 DUP(0FH)      ;Blok za čuvanje promenljivih
```

Poruke o greškama

```
BRACKMSG   DB    'Missing closing bracket!$'
CHARMSG    DB    'Invalid character!$'
VARMSG     DB    'Variable not found!$'
LARGEMSG   DB    'Array index too large!$'
[illegible] DB   'Array indexing error!$'
[illegible] DB   'Array index cannot be zero!$'
```

Tabela karaktera kojima može započeti operand aritmetičkog izraza. Svaki od karaktera je predstavljen svojim ASCII kodom (jedan bajt) iza koga stoji adresa (dva bajta) na kojoj se taj karakter obrađuje. Kraj tabele označen je bajtom 0FFH.

```
SCANTAB    DB    '+'
           DW    OFFSET UNARPLS
           DB    '-'
           DW    OFFSET UNARMNS
           DB    '.'
           DW    OFFSET DECPOINT
           DB    '('
           DW    OFFSET BRACKET
           DB    0FFH
```

Tabela binarnih operacija koje se mogu javiti u okviru aritmetičkog izraza. Svaka operacija je predstavljena svojim ASCII kodom (jedan bajt) iza koga figuriše prioritet i kod operacije, u obliku dvobajtne reči (viši bajt predstavlja prioritet, a niži označava kod operacije u steku kalkulatora). Kraj tabele označen je nulama na mestu ASCII koda, prioriteta i kalkulatorovog koda. Na taj način, ako se iz tabele uzima nepostojeća operacija, biće prikupljena operacija sa kodom i prioritetom nula što je ekvivalentno oznaci kraja aritmetičkog izraza.

```
OPTAB      DB    '*'
           DW    0243H
           DB    '/'
           DW    0240H
           DB    '+'
           DW    0185H
           DB    '-'
           DW    0146H
           DB    '^'
           DW    0487H
           DB    0
           DW    0480H
           DB    00H
           DW    0000H
```

Tabela elementarnih matematičkih funkcija. Svaka stavka tabele sastoji se od nekoliko delova. Prva dva bajta nose informaciju o dužini stavke, iza koga slede ASCII zapis imena funkcije (velika slova abecede) i kao oznaka kraja imena stoji bajt 00H. Poslednji bajt stavke je kod funkcije u steku kalkulatora. Dužina koju sadrže prva dva bajta, meri se od prvog slova tabele funkcije pa do kraja stavke. Kraj tabele označen je nulom (dva bajta) na mestu dužine stavke.

```
FUNCTAB    DB    05H,00H,'ABS',00H,[illegible]
           DB    05H,00H,'INT',00H,[illegible]
           DB    05H,00H,'SQRT',00H,[illegible]
           DB    05H,00H,'SIN',00H,[illegible]
           DB    05H,00H,'COS',00H,[illegible]
           DB    05H,00H,'TAN',00H,[illegible]
           DB    05H,00H,'ASIN',00H,[illegible]
           DB    06H,00H,'ACOS',00H,[illegible]
           DB    06H,00H,'ATAN',00H,[illegible]
           DB    [illegible],'LN',00H,[illegible]
           DB    05H,00H,'EXP',00H,[illegible]
           DB    05H,00H,'SGN',00H,[illegible]
           DB    0000H
```

Program EVALUATE izračunava vrednost zadatog aritmetičkog izraza. Na ulazu, zadati aritmetički izraz mora biti negde u memoriji, u ASCII obliku, a njegova početna adresa treba da je u varijabli CHARADDR. Kraj izraza može biti označen bilo kojim karakterom koji ne predstavlja neku od dozvoljenih binarnih operacija. Rezultat izračunavanja na izlazu smešta se na vrh aritmetičkog steka. Kao operandi aritmetičkog izraza mogu se pojaviti brojevi (u decimalnom obliku), promenljive (iz oblasti VARS) i funkcije (iz tabele FUNCTAB). Između operanada mogu figurisati binarne operacije iz tabele OPTAB. Nivoi zagrada su ograničeni jedino veličinom aritmetičkog steka (blok WORK). Program EVALUATE dozvoljava rekurzivne pozive, jer sve operande i sve operacije koje prolaze, u radu, odlaže na stek.

```
EVALUATE   PROC  NEAR
           XOR   AX,AX          ;Operacija EXIT (prioritet 00H i kod 00H)
           PUSH  AX             ;smešta se na stek operacija.
           CALL  GETCHAR        ;Uzmi prvi karakter aritmetičkog izraza.
OPERAND:   LEA   SI,SCANTAB     ;Testiranje operanda. Da li je to neki od
           CALL  INDEX          ;karaktera iz SCANTAB? Adresa ide u BX.
           JNC   FUNCTEST       ;Skok ako nije iz SCANTAB
           JMP   BX             ;UNARPLS, UNARMNS, DECPOINT ili BRACKET
UNARPLS:   CALL  NEXTCHAR       ;Preskoči predznak plus
           JMP   OPERAND        ;Skok na čitanje operanda.
UNARMNS:   MOV   DX,0507H       ;Prioritet 05H, kod 07H
           PUSH  DX             ;Negativan predznak smesti na stek
           CALL  NEXTCHAR       ;Uzmi sledeći karakter.
           JMP   OPERAND        ;Skok na čitanje operanda.
DECPOINT:  CALL  VALUE          ;Vrednost broja na aritmetički stek
           JMP   OPERATOR       ;Skok na čitanje operacije.
BRACKET:   CALL  NEXTCHAR       ;Uzmi prvi karakter iza zagrade.
           CALL  EVALUATE       ;Vrednost podizraza na aritmetički stek
           CMP   AL,')'         ;Da li je to zatvorena zagrada?
           JNZ   ERRBRACK       ;Ako nije, prijavi grešku.
           CALL  NEXTCHAR       ;Uzmi sledeći karakter
           JMP   OPERATOR       ;Skok na čitanje operacije.
```

## Nastavak sa prethodne strane

```
BRAKRACE:   LEA   DX,BRACKMSG        ;Adresa poruke o grešci.
            MOV   AH,09H             ;Priprema za poziv DOS-a.
            INT   21H                ;Štampanje poruke.
            MOV   AH,4CH             ;Priprema za poziv DOS-a.
            INT   21H                ;Povratak u DOS.
FUNCTEST:   CALL  FUNCTION           ;Da li je to funkcija? Pripremi DX.
            JC    CHARTEST           ;Skok napred ako nije funkcija.
            PUSH  DX                 ;Prioritet i kod funkcije stavi na stek.
            JMP   NOPRAND            ;Skok na čitanje operanda.
CHARTEST:   CALL  ALPHANUM           ;Da li je to slovo ili cifra?
            JNC   BADCHAR            ;Ako nije, prijavi grešku.
            CALL  GETVAL             ;Numerička vrednost na aritmetički stek.
OPERATOR:   CALL  GETCHAR            ;Uzmi tekući karakter.
            LEA   SI,OPTAB           ;Testiranje operacije. Da li je to neki
            CALL  INDEX              ;od kodova iz OPTAB? Formiraj NEW u BX.
OPLOOP:     POP   DX                 ;Skini sa steka operaciju OLD.
            CMP   DH,BH              ;Da li OLD ima veći ili jednak prioritet?
            JAE   TESTEXIT           ;Skok ako ima.
            PUSH  DX                 ;Operacija OLD na stek operacija.
            PUSH  BX                 ;Operacija NEW na stek operacija.
            CALL  NEXTCHAR           ;Uzmi sledeći karakter.
            JMP   OPERAND            ;Skok na čitanje operanda.
TESTEXIT:   AND   DH,DH              ;Da li je to operacija EXIT?
            JNZ   OPMAKE             ;Ako nije, idi napred.
            CALL  GETCHAR            ;Uzmi tekući karakter (iza izraza).
            RET                      ;Povratak. Izračunavanje je završeno.
OPMAKE:     PUSH  BX                 ;Operacija NEW na stek operacija.
            MOV   OPCODE,DL          ;Pošto operacija OLD ima veći ili jednak
            CALL  CALC               ;prioritet u odnosu na NEW, treba je
            DB    06BH  execute      ;izvršiti. Pre poziva kalkulatora,
            DB    0DBH  exitcalc     ;operacija se smešta u OPCODE.
            POP   BX                 ;Skini sa steka operaciju NEW.
            JMP   OPLOOP             ;Natrag u ciklus izvršavanja operacija.
BADCHAR:    LEA   DX,CHARMSG         ;Adresa poruke o grešci.
            MOV   AH,09H             ;Priprema za poziv DOS-a.
            INT   21H                ;Štampanje poruke.
            MOV   AH,4CH             ;Priprema za poziv DOS-a.
            INT   21H                ;Povratak u DOS.
EVALUATE    ENDP
```

Potprogram INDEX obavlja pretraživanje zadate tabele i prikuplja traženi podatak. Svaka stavka tabele mora imati dužinu od tri bajta, pri čemu prvi bajt predstavlja karakter koji se traži, a preostala dva su odgovarajući podaci. Kraj tabele mora biti označen bajtom 00H. Na ulazu u potprogram, registar SI mora sadržati adresu tabele, a AL karakter koji se traži. Ako je pretraga tabele uspela, na izlazu je setovan indikator prenosa, a traženi podatak je u registru BX. Ako pretraga nije uspela, resetovan je indikator prenosa, a BX sadrži podatak iza kraja tabele (pri pretraživanju OPTAB to je takođe nula). U oba slučaja očuvan je AL.

```
INDEX       PROC  NEAR
            MOV   BL,[SI]            ;Uzmi prvi bajt stavke.
            MOV   BX,[SI+1]          ;Uzmi preostala dva bajta.
            ADD   SI,0003H           ;Pripremi Adresu sledeće stavke.
            AND   BL,BL              ;Da li je to bio kraj tabele?
            JNZ   TESTRACE           ;Skok ako nije.
            RET                      ;Povratak. Podatak nije nađen (NC).
TESTRACE:   CMP   BL,AL              ;Da li je to traženi podatak?
            JNE   INDEX              ;Ako nije, skok nazad u petlju.
            STC                      ;Signal: Podatak je pronađen.
            RET                      ;Povratak (CY).
INDEX       ENDP
```

Potprogram FUNCTION proverava da li je zadati operand aritmetičkog izraza neka od elementarnih funkcija iz tabele FUNCTAB, a ako jeste, formira njen kod funkcije. Na ulazu, adresa prvog karaktera zadatog operanda je u CHARADDR. Ako je funkcija prepoznata, na izlazu DL sadrži njen kod u tabeli kalkulatora, a DH njen prioritet (0EH), pri čemu je resetovan indikator prenosa. Tada CHARADDR sadrži adresu odmah iza imena funkcije, a u AL je odgovarajući karakter. Ukoliko funkcija nije prepoznata, na izlazu je setovan indikator prenosa, a CHARADDR i AL sadrže polaznu adresu i odgovarajući ASCII kod.

```
FUNCTION    PROC  NEAR
            LEA   DI,FUNCTAB         ;Adresa tabele sa funkcijama.
            CALL  SEARCH             ;Pronađi datu funkciju (ili kraj tabele).
            MOV   DL,[SI]            ;Uzmi kod funkcije (ili nulu sa kraja).
            MOV   DH,0EH             ;Prioritet funkcije je 0EH.
            MOV   CHARADDR,CX        ;Nova adresa CHARADDR vraćena iz SEARCH.
            CALL  GETCHAR            ;Uzmi tekući karakter.
            RET                      ;Povratak. Indikator prenosa iz SEARCH.
FUNCTION    ENDP
```

Potprogram SEARCH obavlja pretraživanje zadate tabele, sa ciljem da se prepozna zadata reč (ime funkcije, promenljive i slično). Svaka stavka tabele mora imati dva uvodna bajta za oznaku dužine. Iza njih je ime stavke (ime funkcije, osnovne promenljive) praćeno bajtom 0FFH kao oznakom za kraj. Posle ove oznake može doći proizvoljan broj bajtova po potrebi (v. Adresa promenljive, kodovi i prioritet funkcija i slično). Dužina stavke (predstavljena sa prva dva bajta) meri se od prvog slova imena, pa do kraja stavke. Kraj tabele mora se označiti nulom (dva bajta). Na ulazu u potprogram, adresa tabele je u registru DI, a CHARADDR sadrži adresu teksta koji se u tabeli traži. Ako je pretraga uspela, na izlazu je resetovan indikator prenosa, registar SI sadrži adresu parametara koji prate ime stavke u tabeli, a CX sadrži adresu iza prepoznatog teksta (nova vrednost za CHARADDR). Ukoliko pretraga nije uspela, setovan je indikator prenosa, a CX sadrži polaznu adresu CHARADDR. U oba slučaja ostaje očuvan ulazni sadržaj CHARADDR.

```
SEARCH      PROC  NEAR
            CLD                      ;Smer rasta adresa pri radu sa memorijom.
            MOV   CX,CHARADDR        ;Tekuća adresa u aritmetičkom izrazu.
GETENTRY:   MOV   SI,DI              ;Početna adresa stavke.
            LODSW                    ;Dužina stavke u AX.
            MOV   DI,SI              ;Adresa imena stavke.
            AND   AX,AX              ;Da li je to kraj tabele?
            JNZ   MAKENEXT           ;Skok ako nije.
            STC                      ;Signal: Pretraživanje nije uspelo.
            RET                      ;Povratak (CY).
MAKENEXT:   ADD   DI,AX              ;Adresa sledeće stavke.
            MOV   BX,CX              ;Tekuća adresa u aritmetičkom izrazu.
TESTLOOP:   LODSB                    ;Uzmi karakter iz tabele.
            AND   AL,AL              ;Da li je to kraj imena.
            JS    CHECKEND           ;Skok ako jeste.
            MOV   AH,[BX]            ;Uzmi karakter iz aritmetičkog izraza.
            INC   BX                 ;Pripremi Adresu sledećeg karaktera.
            MOV   DL,AH              ;Prenesi karakter u DL, gde će biti
            XOR   DL,1               ;formirana maska: 11011111B ako se radi
            XOR   DL,01000000B       ;o slovu (kodovi preko 40H), a 11111111B
            OR    DL,11011111B       ;za cifre (kodovi ispod 40H).
            AND   AH,DL              ;Pretvori malo slovo u veliko.
            CMP   AL,AH              ;Da li su karakteri isti?
            JE    TESTLOOP           ;Ako jesu, proveravaj dalje.
            JMP   GETENTRY           ;Traži novu stavku.
CHECKEND:   MOV   AL,[BX]            ;Uzmi karakter iza imena.
            CALL  ALPHANUM           ;Da li je to slovo ili cifra?
            JC    GETENTRY           ;Ako jeste, traži novu stavku.
            MOV   CX,BX              ;Nova adresa za CHARADDR.
            RET                      ;Povratak. Pretraživanje je uspelo (NC).
SEARCH      ENDP
```

Potprogram ALPHA ispituje da li zadati karakter predstavlja slovo abecede. Na ulazu registar AL sadrži ASCII kod karaktera koji se ispituje. Ukoliko se radi o slovu (bilo velikom ili malom), na izlazu je setovan indikator prenosa. U protivnom, indikator prenosa je resetovan. U oba slučaja, sadržaj AL ostaje očuvan. Od ostalih registara, koristi se samo AH.

```
ALPHA       PROC  NEAR
            MOV   AH,AL              ;Prenesi karakter u AH.
            AND   AH,11011111B       ;Pretvori malo slovo u veliko.
            CMP   AH,41H             ;Da li je karakter ispod 'A'?
            JAE   TOPTEST            ;Skok ako nije.
            CLC                      ;Signal: Nije slovo.
            RET                      ;Povratak (NC).
TOPTEST:    CMP   AH,5BH             ;Da li je preko 'Z'? Setuj indikator.
            RET                      ;Povratak.
ALPHA       ENDP
```

Potprogram ALPHANUM ispituje da li zadati karakter predstavlja slovo abecede, ili cifru dekadnog sistema. Na ulazu registar AL sadrži ASCII kod karaktera koji se ispituje. Ukoliko se radi o slovu ili cifri, na izlazu je setovan indikator prenosa. U protivnom, indikator prenosa je resetovan. U oba slučaja, sadržaj AL ostaje očuvan. Od ostalih registara, koristi se samo AH.

```
ALPHANUM    PROC  NEAR
            CALL  NUMERIC            ;Da li je to cifra?
            JC    GETALPHA           ;Ako nije, skok napred.
            STC                      ;Signal: Jeste slovo ili cifra.
            RET                      ;Povratak (CY).
GETALPHA:   CALL  ALPHA              ;Da li je slovo? Setuj indikator.
            RET                      ;Povratak.
ALPHANUM    ENDP
```

Potprogram GETVAL uzima iz aritmetičkog izraza sledeću numeričku vrednost (konstantu ili promenljivu) i prenosi je na aritmetički stek. Poziva se kad god se na mestu operanda nađe slovo ili cifra. Tada se [illegible] konstanta prevodi pozivom potprograma VALUE, a promenljiva se obrađuje pomoću LOOKVARS. Međutim, ukoliko zadati operand ne predstavlja ni brojnu konstantu ni ime promenljive, biće prijavljena greška (u LOOKVARS).

```
GETVAL      PROC  NEAR
            CALL  ALPHA              ;Da li je prvi karakter slovo?
            JC    VARIABLE           ;Skok ako jeste (to je promenljiva).
            CALL  VALUE              ;Vrednost broja na aritmetički stek.
            RET                      ;Povratak.
VARIABLE:   CALL  LOOKVARS           ;Pronađi promenljivu u memoriji.
            JNE   VARMOVE            ;Skok ako je obična promenljiva.
            CALL  ARRAY              ;Pronađi zadati element niza.
VARMOVE:    LEA   DI,MEMORY          ;Adresa memorijskog prostora kalkulatora.
            CALL  MOVENUM            ;Prenesi vrednost promenljive u memoriju.
            CALL  CALC               ;Pozovi kalkulator da bi preneo
            DB    0A0H  ;getmem0     ;vrednost promenljive iz memorije
            DB    0DBH  exitcalc     ;na vrh aritmetičkog steka.
            RET                      ;Povratak.
GETVAL      ENDP
```

Potprogram ARRAY pronalazi zadati element višedimenzionalne promenljive (niza ili matrice). Pozivu ovog potprograma obično prethodi LOOKVARS, tako da registar SI na ulazu u ARRAY pokazuje na brojač dimenzija (prva dva bajta iza oznake kraja imena promenljive). Isto tako, CHARADDR mora pokazivati na [illegible] zagradu iza imena promenljive u aritmetičkom izrazu koji se interpretira. Na izlazu, registar SI će sadržati adresu traženog elementa. Međutim, ako se javi neki problem (neodgovarajući broj dimenzija, ili nedozvoljena vrednost indeksa), prijaviće se izveštaj o grešci. Svaki indeks se vrednuje rekurzivnim pozivom EVALUATE, tako da može biti zadat u obliku proizvoljnog aritmetičkog izraza.

```
ARRAY       PROC  NEAR
            CLD                      ;Smer rasta Adresa pri radu sa memorijom.
            LODSW                    ;Prikupi broj dimenzija,
            MOV   CX,AX              ;i prenesi ga u CX.
            XOR   DX,DX              ;Međurezultat DX postavi na nulu.
            JMP   GETINDEX           ;Uđi u ciklus računanja.
GETCOMMA:   CALL  GETCHAR            ;Uzmi karakter iz aritmetičkog izraza.
            CMP   AL,','             ;Da li je to zarez?
            JNE   ERRDIM             ;Ako nije, prijavi grešku.
```

Listing se nastavlja na sledećoj strani

## Nastavak sa prethodne strane

```
GETINDEX:  CALL  NEXTCHAR     ;Uzmi sledeći karakter
           PUSH  CX           ;Sačuvaj brojač dimenzija.
           PUSH  BX           ;Sačuvaj međurezultat.
           PUSH  SI           ;Sačuvaj adresu sledeće dimenzije.
           CALL  EVALUATE     ;Vrednost indeksa na aritmetički stek.
           CALL  GETAX        ;Prenesi vrednost indeksa u AX.
           JC    ERRLARGE     ;Prijavi grešku ako je indeks preveliki.
           DEC   AX           ;Umanji vrednost indeksa za jedinicu.
           JC    ERRZERO      ;Prijavi grešku ako je indeks bio nula.
           MOV   BX,AX        ;Prenesi indeks u BX.
           POP   SI           ;Obnovi adresu dimenzije.
           CLD                ;Smer porasta adresa.
           LODSW              ;Uzmi vrednost dimenzije.
           CMP   BX,AX        ;Da li je indeks manji od dimenzije?
           JNB   ERRLARGE     ;Ako nije, prijavi grešku.
           POP   CX           ;Obnovi međurezultat.
           MUL   CX           ;Pomnoži međurezultat sa dimenzijom.
           ADD   BX,AX        ;Dodaj vrednost indeksa,
           MOV   BX,DX        ;i prenesi rezultat u DX.
           POP   CX           ;Obnovi brojač dimenzija.
           LOOP  GETCOMMA     ;Skok ako ima još dimenzija.
           CALL  GETCHAR      ;Uzmi sledeći karakter.
           CMP   AL,')'       ;Da li je to zatvorena zagrada?
           JNE   ERRSYN       ;Ako nije, prijavi grešku.
           CALL  NEXTCHAR     ;Pređi na sledeći karakter.
           ADD   BX,DX        ;Dobijeni rezultat (redni broj
           ADD   BX,BX        ;elementa niza) pomnoži sa četiri,
           ADD   BX,DX        ;i formiraj adresu elementa.
           RET                ;Povratak.
ERRZERO:   LEA   DX,ERRMSG    ;Adresa poruke o grešci.
           MOV   AH,09H       ;Priprema za poziv DOS-a.
           INT   21H          ;Štampanje poruke.
           MOV   AH,4CH       ;Priprema za poziv DOS-a.
           INT   21H          ;Povratak u DOS.
ERRLARGE:  LEA   DX,LARGEMSG  ;Adresa poruke o grešci.
           MOV   AH,09H       ;Priprema za poziv DOS-a.
           INT   21H          ;Štampanje poruke.
           MOV   AH,4CH       ;Priprema za poziv DOS-a.
           INT   21H          ;Povratak u DOS.
ERRSYN:    LEA   DX,ERRMSG    ;Adresa poruke o grešci.
           MOV   AH,09H       ;Priprema za poziv DOS-a.
           INT   21H          ;Štampanje poruke.
           MOV   AH,4CH       ;Priprema za poziv DOS-a.
           INT   21H          ;Povratak u DOS.
ARRAY      ENDP
```

Potprogram LOOKVARS traži zadatu promenljivu u bloku memorije VARS. Svaka promenljiva ima svoj zapis u okviru bloka, pri čemu prva dva bajta nose informaciju o dužini zapisa. Odmah iza ta dva bajta sledi ime promenljive (proizvoljan niz velikih slova abecede i cifara) praćeno bajtom 00H i tipom promenljive (00H za niz i 01H za običnu promenljivu). Ostatak zapisa zavisi od tipa promenljive. Informacija o dužini, koju nose prva dva bajta zapisa, meri se od prvog slova imena do kraja zapisa. Iza poslednjeg zapisa u bloku VARS mora stajati vrednost 0000H, kao oznaka da nema više promenljivih. Na ulazu u potprogram LOOKVARS varijabla CHARADDR mora sadržati adresu imena promenljive u aritmetičkom izrazu. Ako je pronađena promenljiva sa tim imenom i tipom, na izlazu je vrednost CHARADDR postavljena odmah iza imena promenljive u aritmetičkom izrazu, a SI odmah iza oznake kraja imena u bloku VARS. U protivnom, prijavi se greška.

```
LOOKVARS   PROC  NEAR
           LEA   SI,VARS      ;Adresa bloka promenljivih.
LOOKLOOP:  CALL  SEARCH       ;Pronađi promenljivu sa zadatim imenom.
           JNC   TESTTYPE     ;Skok ako je promenljiva nađena.
           LEA   DX,VARMSG    ;Adresa poruke o grešci.
           MOV   AH,09H       ;Priprema za poziv DOS-a.
           INT   21H          ;Štampanje poruke.
           MOV   AH,4CH       ;Priprema za poziv DOS-a.
           INT   21H          ;Povratak u DOS.
TESTTYPE:  SUB   AL,'('       ;
           SBB   AH,AH        ;vrednost AH zavisno od tipa promenljive
           CMP   AH,AL        ;u izrazu. Ako je tip niz, (zagrada iza
           RCL   AH,1         ;imena) biće AH=0. U protivnom AH=1.
           LODSB              ;Uzmi tip promenljive iz memorije.
           XOR   AL,AH        ;Da li se tipovi podudaraju?
           JNE   LOOKLOOP     ;Ako ne, skok nazad u pretragu.
           MOV   CHARADDR,CX  ;Nova vrednost CHARADDR.
           AND   AL,AL        ;Da li je promenljiva tipa niz?
           RET                ;Povratak.
LOOKVARS   ENDP
```

**Listing 1 Izračunavanje aritmetičkog izraza**

uzimamo samo karakter po karakter iz memorije? Sa gledišta mikroprocesora izraz može započeti jednim od 256 mogućih ASCII karaktera. Šta od tog mnoštva dolazi u obzir?

Numerička vrednost, kao što smo rekli, može biti neki broj, ili neka promenljiva. Ako se radi o broju, onda može početi isključivo nekom od deset cifara dekadnog sistema ili decimalnom tačkom. A ako se radi o promenljivoj, početni karakter može biti samo neko od slova abecede (veliko ili malo).

Bilo koja funkcija takođe može početi samo slovom abecede, uz izuzetak predznaka plus i minus, koji su predstavljeni odgovarajućim ASCII karakterima.

Konačno, podizraz obavezno počinje karakterom „(".

Nakon malog sređivanja svih ovih mogućnosti, doći ćemo do potpunog spiska karaktera kojima može započeti aritmetički izraz, odnosno bilo koji njegov operand. U obzir dolaze sledeći karakteri:

1. Predznak „+"
2. Predznak „–"
3. Decimalna tačka „."
4. Otvorena zagrada „("
5. Slovo ili cifra „A–Z a–z 0–9"

Ako izraz počne nečim drugim, treba prijaviti sintaksnu grešku.

### PROBLEM IZRAČUNAVANJA

Sada, kada imamo definisane sve sastavne elemente aritmetičkog izraza, možemo se pozabaviti problemom samog izračunavanja. Da bi stvar bila jasnija, poći ćemo od jednog jednostavnog primera: *2•3+sqrt(1+x)*. To je prilično prost izraz, koji bi umeo da izračuna i neki osnovac (ukoliko mu, naravno, kažemo koliko iznosi *x*), ali koji u sebi sadrži obilje elemenata na kojima se čovek može lako spotaknuti pri pisanju algoritma.

Svako od nas ima neki svoj „sistem" za računanje, ali bismo u navedenom primeru uglavnom postupali na sledeći način:

Proizvod 2•3 iznosi 6, i to treba upamtiti. Zasebno treba izračunati zbir 1+*x*, pa onda izvaditi koren. Konačno, tome treba dodati onu šesticu.

Prvo što uočavamo je činjenica da uopšte nismo išli redom pri vrednovanju izraza. Posmatrali smo izraz kao jednu celinu i izdvojili u grupe pojedine operande, vodeći računa o *prioritetu operacija* (podizraz u zagradi računali smo nezavisno, pa tek onda tražili koren, dok smo sabiranje između 2•3 i korena obavili poslednje).

Mikroprocesor nema taj dar da vidi ceo izraz odjednom. Umesto toga, on će čitati element po element i operaciju po operaciju, a na nama je da dobrim algoritmom eliminišemo taj nedostatak. Najbolje je, možda, da i sami pokušamo da izračunamo izraz, a da čitamo elemente redom.

Prvi operand je broj dva. Moramo ga negde upamtiti dok ne vidimo šta je iza, a čini nam se da je najpodesnije formirati nekakav stek na koji ćemo slagati pročitane operande. Zapravo, *aritmetički stek*, koji smo tokom ove serije uveli, jeste baš ono što nam treba.

### STEK OPERACIJA

Iza broja dva stoji operacija množenja. I nju treba upamtiti na neki način, pa ćemo zato otvoriti *stek operacija*. Nije trenutno važno kakva je struktura tog steka, odnosno kako se operacija upisuje na njega. O praktičnim detaljima ćemo brinuti kasnije. Koristićemo stek operacija prosto tako što ćemo, po potrebi, slagati na njega pročitane operacije, da bismo ih kasnije skidali i izvršavali. Jedino će biti podesno pre prve stvarne operacije ubaciti na dno steka nekakvu oznaku početka, u smislu „nema više". Na taj način, kada budemo skidali operacije, imaćemo mogućnost kontrole kraja procesa. Uslovno ćemo i oznaku početka izraza zvati „operacijom", što one donekle i jeste.

Sada, kada je operacija množenja otišla na stek, možemo čitati izraz dalje. Sledi broj tri. Ne smemo, međutim, množiti 2•3, jer još ne znamo šta je iza trojke. Tek kada budemo sigurni da je tamo operacija nižeg prioriteta, moći ćemo da obavimo množenje. Dakle, i trojku smeštamo na aritmetički stek, odmah iznad dvojke.

Iza trojke je znak sabiranja. Sa steka operacija čitamo da je prethodna operacija bila množenje, a pošto se radi o većem prioritetu, obavljamo množenje dva poslednja broja na aritmetičkom steku (baš zgodno, jer smo sve binarne operacije kalkulatora napravili tako da operišu sa brojevima u vrhu steka). Posle množenja na steku ostaje šestica, a znak sabiranja stavljamo na stek operacija, gde je trenutno „usamljen" (množenje smo skinuli sa steka i „utrošili").

Dolazi funkcija kvadratnog korena. Nju stavljamo na stek, iznad sabiranja.

### OD SAMOG POČETKA

Otvorena zagrada je znak da sledi podizraz koji treba zasebno računati. Jednostavno, pravićemo se kao da ništa do sada nismo radili i krenućemo od samog početka: oznaku početka izraza ubacujemo na stek operacija i krećemo sa čitanjem podizraza. Naravno, aritmetički stek i stek operacija čuvaju i dalje „tragove" naše prethodne aktivnosti.

Prvi operand podizraza (jedinicu) stavljamo na aritmetički stek, znak sabiranja na stek operacija, a vrednost promenljive *x* ponovo na aritmetički stek. Stanje na oba steka bi sada bilo sledeće: aritmetički stek (od dna prema vrhu) sadrži tri broja (šesticu, jedinicu i vrednost *x*), dok stek operacija sadrži pet operacija (oznaku početka, sabiranje, koren, drugu oznaku početka i sabiranje).

Iza promenljive *x* ne stoji nikakva operacija, što je znak da smo stigli do kraja izraza (tačnije podizraza koji trenutno računamo). To znači da treba sada redom skidati sa steka operaciju po operaciju i izvršavati ih sve do dna (odnosno do oznake početka).

**NASTAVLJA SE NA STRANI 82**

# SEKS, TEST I RAČUNARI

**Sve dok nedavno čuveni Kinsey institut nije objavio svoja istraživanja o seksualnom ponašanju i znanju Amerikanaca na gala priredbi u Njujorku, mislila sam da su testovi prilično zabavna i bezopasna stvar. Ako ne za one koji ih popunjavaju, onda bar za one koji ih prave i distribuiraju. Ali posle opširnog pisanja u skoro celokupnoj Američkoj štampi, nedvosmisleno se mogla zaključiti nekoliko izuzetno zanimljivih stvari.**

SEKA R. JOVIĆ

Prvo, da autori testa nisu znali tačan odgovor na sva pitanja koja su postavili. Recimo, nije tačno, kako su oni tvrdili, da su problemi erekcije najčešće posledica fizičkih smetnji, naprotiv, oni su najčešće uslovljeni psihičkim smetnjama. (To se moglo proveriti u svakom udžbeniku koji tretira doličnu problematiku.) Zatim, da je besmisleno tvrditi da 60% žena ne zna šta većina žena misli o preferenciji žena u odnosu na veličinu muškog polnog organa.

Drugo, većina pitanja koja su postavili u testu — koji je dotični institut koštao zamašnu količinu dolara — su potpuno nevažna. Istina, neka saznanja su interesantna, ali pored tolikih urgentnih problema pitanja tipa „kad većina ljudi jede voće za vreme ručka, pre ili posle deserta" su smešna i žalosna istovremeno.

I naravno, saznalo se da su profesori na Indijana univerzitetu, uz koga je Kinsey institut smešten, već duže vremena zahtevaju da se zatvori jer nije na „dovoljno visokom naučnom nivou". A autorica testa, koja je direktor instituta, je otpuštena i samo zahvaljujući pravnim smicalicama zadržava svoju fotelju. I tako dalje i tako dalje.

Iako pretpostavljam da bar jedan deo čitalaca sada nestrpljivo očekuje Kinsey sex test, ovog puta moraće da se zadovolje testom nešto drugačijeg sadržaja.

Tema ovog testa je jednako kontraverzna kao i kod Kinsey testa. To su, naravno, računari ili još preciznije računarstvo.

Koliko stvarno znate o računarima — o tome kako ih koristiti i o tome kako ih projektovati? (Na žalost, pitanja o tome kako ih preprodavati nisu zastupljena.) To ćete jednostavno utvrditi ako se podvrgnete GRE testu iz računarstva (u originalu Computer Science). Ipak, pre nego što se prijavite na adresu (i pri tom potrošite četrdesetak dolara)

Graduate Record Examination Program
CN 6000
Princeton, NJ 08541—6000
USA

pogledajte šta možete da očekujete na testu. Naravno, ako vas test nervira, ili su rezultati nešto drugačiji nego što ste očekivali, nemojte se uzbuđivati. Takvi i slični testovi su za druge, a ne za vas, prevejanog i iskusnog čoveka od (računarske) prakse. A možda će se uskoro otkriti da su odgovori koje autori testa smatraju tačnim, u stvari samo njihova lokalna zabluda. Ili su pitanja totalno irelevantna. Ako pak zasijate, onda je to dobar pokazatelj da su vam vrata računarskih institucija širom sveta otvorena.

Pitanja dolaze iz pet globalnih oblasti: programski sistemi i metodologija, organizacija i arhitektura računara, teorija, računarska matematika i specijalne oblasti.

Iz **programskih jezika i metodologije** pitanja dolaze u vezi sa organizacijom podataka (apstraktni tipovi podataka, implementacija tipova podataka, kao što su pointeri), sa organizacijom kontrole programa (iteracije i rekurzije, funkcije i procedure, konkurentni procesi, međuprocesorska komunikacija i sinhronizacija), sa programskim jezicima (aplikativni i proceduralni jezici, prenošenje podataka i izračunavanje izraza) i sistemima (kompajlera, operativnih sistema, dodeljivanje resursa i protekcija i zaštita).

Za **arhitekturu i organizaciju računara** tipična pitanja su logičkog projektovanja procesora i kontrolnih jedinica, organizacije memorije (cache, primarna i sekundarna memorija, virtuelna memorija) i organizacije ulaza/izlaza.

**Teorija** obuhvata automate i teoriju jezika (regularni jezici, konačni automati, konteksno slobodni jezici, specijalne klase konteksno slobodnih gramatika, Tjuringove mašine, gornje i donje granice i analiza algoritama (i sasvim retko NP kompletni zadaci).

**Računarska matematika** ima dve podoblasti: diskretne strukture i numeričku matematiku. U specijalne oblasti spadaju veštačka inteligencija, baze podataka, modeliranje i simulacija, računarska grafika, računarske telekomunikacije, ali kako su ove oblasti zastupljene sa manje od 5% pitanja na njih ne treba stavljati poseban akcenat pri spremanju testa. Sve prethodne oblasti zastupljene sa po oko 20% pitanja, s tim što na programske jezike i metodologiju otpada 35% pitanja.

Generalna filozofija testa je da pitanja budu prilično jednostavna, pa se na njih može odgovoriti iako niste previše upoznati sa dotičnom oblašću. Na dobar deo pitanja se može odgovoriti koristeći znanja iz elementarne računarske pismenosti i ne suviše komplikovanog razmišljanja. Posebna pogodnost je što je dovoljno odgovoriti (naravno, tačno) na 75% pitanja i postići jako dobar rezultat.

I na kraju, kao poslasticu, navodimo skraćenu verziju od 40 pitanja. Vreme rada je 80 minuta. Svaki tačan odgovor vredi 1 poen, netačan $\frac{-1}{4}$. Na kraju pogledajte rezultate, saberite (i oduzmite) svoje poene i pogledajte svoju računarsku snagu.

1 U jeziku u kome su operacije asocijativne sdesna nalevo, umesto sleva nadesno (npr. a+b+c=a+(b+c), vrednost izraza 7−(16/(3+1) 2)−4 je
(A) −1 (B) 1 (C) 3 (D) 7 (E) 9

2 Koja je od sledećih aproksimacija odnosa broja neterminalnih čvorova u odnosu na ukupan broj čvorova u kompletnom K arnom drvetu dubine $N$?
(A) $\frac{1}{N}$ $N-\frac{1}{N}$ (C) $\frac{1}{K}$ (D) $K-\frac{1}{K}$ (E) $\log_{10}\frac{1}{N}$

3
```
if A>B then
   V [I] = F (I)
else
   if B>C then
      V [I] = G (I)
```
Pretpostavimo da je, u proseku, $A>B$ u 75% slučajeva i $B>C$ u 25% slučajeva. Ako se gore naveden deo programa izvrši 10,000 puta, koliko puta se očekuje da se funkcije $F$ i $G$ izvrše?
(A) $F$ 2 500 $G$ 18 750 (B) $F$ 7,500 $G$ 625
(C) $F$ 7 500 $G$ 1,875 (D) $F$ 7,500 $G$ 2 500
(0) $F$ 9,375, $G$ 625

4 U sledećoj funkciji $X$ se prenosi referencom na lokaciji (by reference), a $Y$ po vrednosti.
```
function P(var X: integer, Y: integer): integer
begin
   K = 3,
   L = 5,
   P = X+Y
end;
```
Ako se funkcija $P$ poziva sledećim delom programa
```
   K = 1,
   L = 1,
   Z = P (K, L),
```
tada će vrednost $Z$ biti jednaka
(A) 2 (B) 3 (C) 4 (D) 6 (E) 8

5 U balansiranom ternarnom brojnom sistemu baza sistema je 3, a cifre su 0, 1 i −1 (označena kao $\bar{1}$). Balansirani ternarni broj jednak decimalnom broju $35\frac{2}{9}$ je
(A) $11\bar{1}1.01$ (B) $11\bar{1}1.\bar{1}1$ (C) $110\bar{1}.\bar{1}1$
(D) $110\bar{1}.01$ (E) $110\bar{1}.1\bar{1}$

6 Regularni izraz koji označava sve niske sastavljene od cifara 0 i 1, i koje imaju najmanje dve uzastopne cifre 1 je
(A) (0+10)* 11 (10+0)*
(B) (0+10)* 11 (0+1)*
(C) (0+1)* 10 1(0+1)*
(D) 0*11 (0+10)*(0+1)*
(E) (0+11)*

7 Ako je $f$ definisano kao
```
function f (x: integer): integer,
begin
   if x=1 then
      f := 0
   else
      f := x*f (x−1)+x*x
end,
```
Tada će vrednost $f$ (4) biti
(A) 29 (B) 50 (C) 53 (D) 100 (E) 48

8 Koliki je broj različitih terminalnih niski dužine 5 koji je moguće generisati konteksno slobodnom gramatikom?
$S \rightarrow A0B$
$A \rightarrow BB \mid 0$

$B \to AA \mid 1$

(A) 4 (B) 5 (C) 6 (D) 7 (E) 8

9 Broj jedinica u binarnoj predstavi broja $13 \cdot 16^3 + 11 \cdot 16^2 + 9 \cdot 16 + 3$ je tačno

(A) 7 (B) 8 (C) 9 (D) 10 (E) 12?

10 Koji od sledećih algoritama za sortiranje ima u prosečnom i najgorem slučaju vreme izvršavanja $O(n \log n)$?

(A) Bubble sort (B) Insertion sort
(C) Merge sort (D) Quicksort
(E) Selection sort

11 U određenom operativnom sistemu, vreme potrebno za iščitavanje datoteke sastoji se od 4 komponente koje se ne preklapaju

vreme potrebno za pretragu diska — 25 msec
vreme potrebno za zaustavljanje diska — 8 msec
vreme za transfer sa diska — 1 msec za 1,000 bajtova
režijsko vreme operativnog sistema — 1 msec za 1 000 bajtova+10 msec

U verziji 1 datoteke se iščitavaju u blokovima od po 1 000 bajtova U verziji 2 datoteka se očitava u blokovima od po 4 000 bajtova Vreme potrebno da se iščita vrlo velika datoteka u verziji 2 u odnosu na vreme da se ista datoteka iščita u verziji 1 je aproksimativno

(A) 1 4 (B) 1 3 5 (C) 1 1
(D) 1 1 1 (E) 2 7 1

12 Ponekad se uz objektni moduo koji se generiše kompajlerom dodaje (iz labele simbola) tabela svih imena iz izvornog programa sa njihovim adresama Ova informacija ce se najverovatnije koristiti

(A) za pomoć pri ispravljanju programa
(B) za povećavanje brzine izvršavanja programa,
(C) za redukciju prostora simbola labele koje zahteva kompajler,
(D) da bi se reklo punioci gde koja promenljiva pripada,
(E) da bi se reklo operativnom sistemu kako da poziva promenljive,

13

| A | B | C | f |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | x |
| 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | x |

U gornjoj tabeli $x$ označava da promenljiva može da uzme bilo koju vrednost Koja od sledećih rečenica tačno opisuje odnos između minimalne sume i minimalnog proizvoda od $f$?

(A) One su logički ekvivalentne po definiciji,
(B) One su logički ekvivalentne zato što se $x$ koristi na isti način,
(C) One su logički ekvivalentne zato što $x$ ne utiče na odnos ovih veličina
(D) One nisu logički ekvivalentne po definiciji,
(E) One nisu logički ekvivalentne zato što se $x$ koristi na različite načine

14 Binarna realizacija celih brojeva definisana je na sledeći način $R = \{ (x,y) \mid |y-x| \leq 1 \}$ i ima sledeće osobine

I Refleksivnost
II Simetričnost
III Tranzitivnost

(A) nijednu (B) I i II samo (C) I i III samo
(E) I, II i III

15 Sledeći deo programa je napisan na asemblerskom jeziku za jednoadresni računar sa jednim registrom za akumulaciju

```
LOAD  B
MULT  C
STORE T1
ADD   A
STORE T2
MULT  T2
ADD   T1
STORE Z
```

Koji je aritmetički izraz implementiran gore navedenim delom programa?

(A) $z = t_1(bc + a) + t_2$
(B) $z = (a + bc)^2 + bc$
(C) $z = 2bc + a^2$
(D) $z = (a + bc) + bc$
(E) $z = a + 2bc$

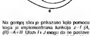
16 Jezik $\{ww \mid w \in (0+1)^*\}$

(A) ne prihvata ni jedna Turingova mašina,
(B) prihvataju neke Tjuringove mašine, ali ne i pushdown automati
(C) prihvataju neki pushdown automati ali ne konteksno slobodni,
(D) je konteksno slobodan, ali nije regularan,
(E) regularan je

17 Broj načina na koji se iz jugozapadnog ugla celobrojne pravougaone mreže $A(0,0)$ može stići do severoistočnog ugla $B(n,n)$, ako se u svakom koraku moramo kretati za 1 liniju severno ili istočno

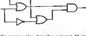
(A) $2^{2n}$ (B) $\binom{2n}{n}$ (C) $n!$ (D) $n^2$

(E) $n(n+1)/2$




18 Na gornjoj slici je prikazano kolo pomoću koga je implementirana funkcija $z = f(A, B) = A + B$ Ulazi $I$ i $J$ mogu da se postave na vrednosti iz skupa $\{0, 1, B, \bar{B}\}$ Koje vrednosti treba da se uzmu za $I$ i $J$?

(A) $I = 0$ $J = B$ (B) $I = 1$, $J = B$ (C) $I = B$ $J = 1$
(D) $I = \bar{B}$, $J = 0$ (E) $I = 1$, $J = \bar{B}$




19 Na gornjoj slici, konačni automat $M$ ima startno stanje $A$ i cilj $C$ Koji od sledećih izraza označava skup reči koje prihvata automat $M$?

(A) $10^*$ (B) $(10)^*$ (C) $10^*1^*0$
(D) $1(01)^*1$ (E) $(01)^*01$

20 Koji od sledećih regularnih izraza opisuje nisku sledeće gramatike (sa korenom $S$)?

$S \to Ax \mid By$
$A \to y \mid Ay$
$B \to x \mid y$

(A) $y^*x + xy + y$ (B) $y^*x + x + yy$
(C) $yy^*x + xy + y$ (D) $y^*x + xy + yy$
(E) $yy^*x + xy + yy$

21 Jednopovršinski disk ima sledeće karakteristike

broj staza na disku — 35
broj sektora na stazi — 10
brzina prenosa — 250 000 bita u sekundi
rotaciona brzina — 300 obrta u minuti
jedan bajt je 8 bita

Ako pretpostavimo da nema praznina u formatu i da se ne koriste specijalni formati, nominalni kapacitet diska u bajtovima je

(A) 6,250 (B) 29,267 (C) 218,750
(D) 1,750 000 (E) nijedna od navedenih vrednosti

22 U mikroprogramiranoj kontrolnoj jedinici računara ZVRK ZVRK izlaz kontrole se smešta u kontrolni registar Kontrolni registar može da adresira tri izvora: (I) izlaz mikroprogramiranog brojača za sekvencijalan postup mikroinstrukcijama, (II) polje kontrolnog registra za grananje mikroinstrukcije i (III) dodatni ulaz na koji su spojeni signali ostalih delova ZVRK ZVRKa Dodatni ulaz će se NAJMANJE verovatno koristiti za signale sa

(A) CPU opcode instrukcije,
(B) CPU adrese instrukcije
(C) CPU bita moda adrese instrukcije,
(D) aritmetičko logičke jedinice,
(E) prekida

23 Neka je < S > sintaksna kategorija definisana Bakus Naurovom formom

< S > ::= rl | r < S > l | < S >< S >

Koji od sledećih niski mogu da se generišu iz < S >?

I rrlrl
II rlrrrll
III rrlrlrlrlrl

(A) samo I (B) samo II (C) samo III
(D) I i III (E) II i III

24 Šestobitni brojevi se mogu predstaviti u drugom komplementu, prvom komplementu ili znakom i apsolutnom vrednošću U kojim vidovima predstavljanja dolazi do prekoračenja opsega (overflowa) pri sabiranju brojeva 011000 i 011000?

(A) u drugom komplementu
(B) znaku i apsolutnoj vrednosti i prvom komplementu,
(C) drugom i prvom komplementu,
(D) drugom komplementu i znaku i apsolutnoj vrednosti,
(E) u sve tri predstave

25 U sledećem segmentu programa pretpostavlja se da je pre početka izvršenja programa $m \leq n$ i $a[m-1] \leq w \leq a[n+1]$

```
i := m-1,
j := n+1,
while i<j do
begin
   repeat
     i := i+1
until a[i] >= w
   repeat
     j := j-1
until a[j] <= w,
if i<j then
   begin
     Temp := a[i],
     a[i] := a[j],
     a[j] := Temp,
   end
end
```

Za velike vrednosti $n-m$ koja je najbolja aproksimacija broja izvršavanja dodeljivanja vrednosti $i := i+1$ i $j := j-1$?

(A) $n-m$ (B) $2(n-m)$ (C) $(n-m)^2$ (D) $(n-m)^2/2$
(E) $(n-m)(n-m+1)/2$

26 Ako su $A$ i $B$ brojevi u pokretnom zarezu sa 10 bitnom mantisom, ako je $A=1.0$ i $B=2^{-10}$ koji je (su) od sledećih tvrđenja tačno (tačna)?

I $A = A + B$
II $B = A + B$
III $A + (B) + B = A + B + (B)$

(A) samo I (B) samo III (C) I i II
(D) II i III (E) sva tri

27 Standardan način za generisanje slučajnih brojeva pomoću računara je metoda linearnih kongruentnih generatora

izabrati $x_0$ (ceo broj)
$x_{i+1} = (ax_i + b)$ **mod** $m$ za $i = 0, 1, 2, \ldots$

pri čemu su $a$, $b$ i $m$ pozitivni celi brojevi. Period $P$ niza generisanog ovakvim generatorom je najmanji pozitivni ceo broj za koga je $x_{i+P} = x_i$ za svako $i$. Koja je od sledećih tvrdnji tačna?

(A) $P \le m$ za svako $a, b, x_0$
(B) $P$ može da bude jednako $2^m$ za određene vrednosti $a, b, x_0$
(C) Za dato $m$ i $x_0$, $P$ je nezavisno od $a$ i $b$
(D) Za sve kombinacije $a, b$ i $m$, $P$ je nezavisno od $x_0$
(E) Nijedna od gornjih tvrdnji nije tačna

28 Konstrukcija **cobegin** *Statement1*; *Statement2* **coend** označava da se *Statement1* i *Statement2* izvršavaju paralelno

```
x := 0;
y := 0;
cobegin
 begin
  x := 1;
  y := y+x
 end;
 begin
  y := 3;
  x := x+3
 end;
coend
```

Koje su od sledećih moguće vrednosti promenljivih posle izvršenja gornjeg segmenta?

I $x=1$, $y=2$
II $x=1$, $y=3$
III $x=4$, $y=6$

(A) samo I (B) samo I i II (C) samo I i III
(D) samo II i III (E) sve tri

29 Za $x \ge 0$, $y \ge 0$ definišimo $A(x,y)$ sa

$A(0,y) = y+1$
$A(x+1,0) = A(x,1)$
$A(x+1,y+1) = A(x,A(x+1,y))$

Za sve pozitivne vrednosti $y$, $A(1,y)$ je

(A) 2 (B) $y+1$ (C) $y+2$ (D) $2y+3$
(E) nijedna od navedenih

30 Koje od sledećih dodeljivanja ima osobinu ako je dodeljivanje tačno pre izvršenja programskog dela $z := z*a$; $y := y-1$ tačno je i posle?

(A) $a*y = z$ (B) $z = a^y$ (C) $a^y = z*a^y$
(D) $y > 0 \wedge z = a^y$ (E) nijedno od navedenih

31 Koji od sledećih regularnih izraza je ekvivalentan izrazu (a*+(B)*(c+(D)?

(A) a*(c+(d) + b(c+(d)
(B) a*(c+d)*+b(c+d)*
(C) a*(c+d)+b*(c+(d)
(D) (a+b)*c+(a+b)*d
(E) (a*+b)c+(a*+b)d

32 U informacionom sistemu svaki zapis ima petobitni ključ. Kao odgovor na pitanje čiji je petobitni ključ $q$ sistem daje kao odgovor sve zapise koji imaju Hemingovo rastojanje najviše za 1 različito od $q$. Pretpostavimo da su ključevi zapisa u sistemu

00000
00011
01101
10100
11111

Kao odgovor na proizvoljan upit $q$ koliko se minimalno i maksimalno zapisa može dobiti kao odgovor sistema

| | Minimum | Maksimum |
|---|---|---|
| (A) | 0 | 2 |
| (B) | 0 | 1 |
| (C) | 1 | 2 |
| (D) | 0 | 3 |
| (E) | 1 | 5 |

33 Neka je $b_n b_{n-1} \ldots b_0$ binarna predstava celog broja $b$. Broj 3 je delilac broja $b$ ako i samo ako je

(A) $b_1 = b_0 = 1$
(B) zbir binarnih cifara $b$ deljiv sa 9
(C) zbir binarnih cifara $b$ deljiv sa 3 ili ne i sa 9
(D) ako je alternativna suma $b_0 - b_1 + b_2 - \ldots$ nula
(E) ako je alternativna suma $b_0 - b_1 + b_2 - \ldots$ deljiva sa 3

34 8 zadataka je potrebno obraditi na dva identična procesora. Vremena u kom su poslovi pristupni procesorima i dužina njihove obrade je data u tabeli

| zadatak | pristupan | vreme obrade |
|---|---|---|
| A | 0 | 6 |
| B | 0 | 2 |
| C | 0 | 3 |
| D | 2 | 5 |
| E | 3 | 4 |
| F | 5 | 1 |
| G | 7 | 3 |
| H | 9 | 6 |

U kom trenutku se svi poslovi mogu najranije završiti?

(A) 14 (B) 15 (C) 16 (D) 17 (E) 18

35 Koliko je minimalno kašnjenje za izvršenje svih poslova u prethodnom zadatku?

(A) 1 (B) 1.25 (C) 1.38 (D) 1.5 (E) 1.75

36 Visoko balansirano stablo je stablo kod koga se visina levih i desnih potomaka razlikuje za najviše 1. Koji od sledećih iskaza važe za ovo stablo?

I u najnepovoljnijem slučaju vreme pretrage je logaritamsko u odnosu na broj čvorova
II u srednjem slučaju vreme pretrage je logaritamsko u odnosu na broj čvorova
III u najpovoljnijem slučaju vreme pretrage je proporcijalno visini drveta
IV visina drveta je logaritam broja čvorova

(A) I i III samo (B) II i III samo
(C) II i IV samo (D) I, II i III (E) I, III i IV

37

```
for Pass := 1 to N-1 do
begin
  for Item := 1 to N - Pass do
  begin
    if List [Item] < List [Item+1] then
    begin
      Temp := List [Item];
      List [Item] := List [Item+1];
      List [Item+1] := Temp
    end
  end
  (*1*)
end
```

Broj poređenja prema ovom algoritmu su u odnosu na N broju elemenata u listi kao

(A) $O(N)$ (B) $O(2(N-1))$ (C) $O(N\log N)$
(D) $O(N^2)$ (E) $O(N^3)$

38 Sledeći program je napisan na jeziku koji ima sintaksu paskala, ali prenos parametara nije ograničen na *call by value* i *call by reference* kao paskalu

```
program Param (input, output);
var
  a, b: integer;
  procedure p(x, y: interger);
  begin
    x := x+2;
    a := x*y;
    x := x+1
  end;
begin
  a := 1;
  b := 2;
  p(a, b);
  writeln(a)
end.
```

Pod kojim od sledećih načina prenosa parametara bi izlazni rezultat bio 4?

(A) call by value
(B) call by reference
(C) call by name
(D) call by value result
(E) njedan od navedenih

39 Uzmimo u obzir sledeći algoritam za $n>0$

```
const
  n = <neki pozitivan ceo broj>;
var
  a, b, p, Sum: integer;
begin
  a := 0; b := n; Sum := 0;
  readln(p);
  while a<n and b>0 do
  begin
   if p = 0 then
   begin
     a := a+1;
     Sum := Sum+1
   end
   else
   begin
     a := a - 1;
     b := b - 1;
     Sum := Sum-1
   end;
   Modify(p) {*p called by reference    *}
             {*Modify ne utiče ni na jednu *}
             {*više promenljivu           *}
  end
end.
```

Koji rezultat najbolje određuje najveći mogući rezultat promenljive *Sum* kada algoritam završi?

(A) $3n$ (B) $4n$ (C) $5n$ (D) $2n^2$ (E) $3n^2$

40 Koji od sledećih iskaza mora biti tačan

I $\lceil x \rceil = \lfloor x \rfloor$ ako i samo ako je $x$ integer
II $\lfloor x \rfloor + 1 = \lceil x \rceil$ ako i samo ako $x$ nije integer
III $\lceil x \rceil \lceil y \rceil = \lceil xy \rceil$ za svako $x$ i $y$
IV $\lfloor x \rfloor = -\lceil -x \rceil$ za svako $x$

(A) IV samo (B) I i IV samo
(C) I, II i III samo (D) I, II i IV samo
(E) I, II, III i IV

NASTAVAK NA STR. 82

## SPRINTF

Izuzetna fleksibilnost pri formatiranju ispisa koju nudi funkcija **printf** u mnogim programima se dovodi u pitanje jer je njen ispis usmeren na standardni izlaz i koristi usluge DOS-a. Kao što je poznato, brzina koja se ostvaruje ovakvim načinom ispisivanja nije baš pogodna za programe koji računaju na efektniji izgled ekrana popunjen rasterima i ramovima. Rešenje se obično traži u BIOS funkcijama ili direktnom gađanju video memorije, ali se time odričemo formatiranog ispisa i svih lepih opcija **printf** funkcije. Idealno bi zato bilo na neki način preusmeriti ispis **printf** funkcije na neku našu rutinu koja će ispis obaviti brže.

Jedno rešenje je da koristimo funkciju **sprintf** koja svoj ispis vrši u string, na primer:

```
sprintf(buf,"%04d %-16s %ld\n",rb,name,val);
biosprint(buf);
```

Slika 1

pri čemu smo sa **biosprint** označili poziv neke jednostavne funkcije koju treba samo da sadržaj stringa što brže „saspe" na ekran. Ovakvo rešenje nije ipak elegantno jer su za svaki ispis potrebni pozivi dve funkcije. Pravo rešenje nudi funkcija **vsprintf**.

Pošto naša funkcija treba po svemu da bude identična sa **printf** (osim, naravno, po brzini ispisa), potrebno je da je deklarišemo kao funkciju sa promenljivim brojem argumenata. Sve neophodne deklaracije se nalaze u zaglavlju **stdarg.h** koje treba uključiti u program. Funkcija **bprintf** (od bios printf) bi izgledala ovako:

```
#include <stdarg.h>
void bprintf(char *format, ...)
{
    static char b[256];
    va_list arg_ptr;
    va_start(arg_ptr, format);
    vsprintf(b, format, arg_ptr);
    biosprint(b);
}
```

Slika 2

**va_start** je makro definisan u **stdarg.h** i ima zadatak da u naš pointer (u primeru **arg_ptr**) postavi adresu prvog parametra na steku posle **format** stringa. Nakon toga se ova vrednost prosleđuje funkciji **vsprintf** koja se, dakle, od obične **sprintf** razlikuje jedino po tome što ne očekuje da se na njenom steku nalaze redom parametri, već prima pointer na stek „tuđe" funkcije (**bprintf**). Nakon što završi svoj posao, u stringu **b** će se naći formatizovan ispis koji potom funkcijom **biosprint** jednostavno ispisujemo na ekran.

U primeru vam verovatno nije promakla i jedna konstanta, 256, kojom sam odredio najveću dužinu stringa koja se ispisuje. Zbog toga što C nema nikakve provere prekoračenja granice niza, vrednost 256 kao dimenziju niza **b** sam izabrao kao „pristojno veliku", pa jedino što nam ostaje je da se nadamo da neki ispis neće preleteti preko ove granice. Da li je moguće ipak na neki način nadu zameniti sigurnošću? Odgovor je ne.

Najveći broj ispisa koji se pojavljuje u programu je, ipak, ograničen na jedan red (80 znakova) pa bi se zaista moglo tvrditi da, recimo

```
bprintf("%04d %-40s %-20s",rb,ime,grad);
```

Slika 3



neće preći 66 mesta, koliko bi trebalo da bude po zadatom formatu, a kamoli granicu od 256 znakova. Ipak, pri „probijanju" zadatih formata (dakle ako je recimo string **ime** duži od 40 znakova ili **grad** od 20), ispis se nesmetano nastavlja, pa granica od 66 može biti prekoračena. Da li možemo da tvrdimo da smo u ovom slučaju sigurni sa nizom od 256 znakova, da li je stvarno moguće da se baš toliko prekorače očekivane dužine stringova? U normalnim okolnostima, pošto i stringovi **ime** i **grad** imaju svoju dimenziju koja očigledno može biti prekoračena samo nekorektnim ponašanjem programa, odgovor bi zaista mogao biti – „nema šanse"!

Šanse, na žalost, ima, jer nisu stvari uvek ovako jednostavne kao u primeru. Možete mi verovati na reč da će vam se kad-tad desiti da zbog ovog niza program misteriozno „padne" i da je praktično nebitno koju ste dimenziju izabrali (da se Marfi bavio ovim problemom, verovatno bi ga definisao pravilom da je broj krahova programa zbog prekoračenja granice niza direktno proporcionalan veličini niza!)

Na žalost, rešenje ovog problema nije jednostavno, ili bar meni nije poznat jednostavan način osim pisanja sopstvene **vsprintf** funkcije. Unutar **vsprintf** se obično koristi pomoćna funkcija (u mnogim bibliotekama se naziva baš **output**) koja se poziva kada jedan znak treba smestiti u niz. Ona je idealno mesto da se uz pomoć nekog globalnog parametra ograniči ispis na zadatu vrednost, ali je mala analiza MSC biblioteke pokazala da, iako je funkcija stvarno tu, njena konstrukcija nije jednostavna, pa bi njena zamena bila dovoljno komplikovana da se od celog posla odustane. Imate li vi neko kratko rešenje?

### SAVETOVALIŠTE ZA C

Namena ove rubrike je da se na nesistematizovan način, onako kako se ukaže prilika, bavi važnim i manje važnim problemima pri upotrebi C jezika. Iskustva i pitanja će se odnositi ne samo na programski jezik već i na sve alate koji se u procesu pisanja koriste – editore, kompajlere, linkere, dibagere... jednom rečju, na sve ono na šta možete naleteti pišući jedan C program. S obzirom da isključivo koristim *Microsoft C* kompajler, najveći broj iskustava će se odnositi na njega što, nadam se, neće umanjiti korist od ovih tekstova.

Ova rubrika je, ujedno, i prilika da svojim pitanjima, savetima i iskustvima obogatite njen sadržaj. Kontakt sa mnom možete ostvariti putem pisma na adresu Redakcije ili privatnom porukom na *Sezamu* za **zzivotic**.

## MATEMATIČKA BIBLIOTEKA

Često se dešava da vaš prvi sudar sa funkcijama koje operišu nad realnim promenljivima (tipa **float** i **double**) bude pozdravljen **run-time** porukom o grešci

```
run-time error R6002
- floating point support not loaded
```

Slika 4

Ako vas recimo interesuje u kom formatu se čuvaju realne promenljive, možete pokušati sledećim programom:

```
main()
{
    float r;
    int i;
    unsigned char *pr;
    printf("Upisi neki broj: ");
    scanf("%f",&r);
    pr = (unsigned char *)&r;
    for(i=0;i<sizeof(float);++i)
        printf("%4d",*pr++);
}
```

Slika 5

Od ovog programa, sem pomenute run-time greške, neće biti nikakve koristi. Problem je u tome što je podrška za rad sa realnim promenljivima prilično glomazna i svi kompajleri se trude da je ubace u program samo ako je zaista neophodna. Odluka se donosi na osnovu pojavljivanja floating point (FP) operacija u programu, ali je, na žalost, u ovom primeru kompajler doveden u neprijatnu situaciju – jedino mesto na kome se pojavljuje bilo kakva realna promenljiva je unutar **scanf** funkcije, koja po definiciji koristi promenljiv broj argumenata, pa njene argumente kompajler ni ne proverava. Neki kompajleri se odlučuju da FP biblioteku ubace čim se pojavi funkcija **scanf** ili **printf** (kao što to izgleda radi *Turbo C*), ali je MSC u ovom slučaju i pored svog „pametnijeg" algoritma doneo pogrešnu odluku.

Ipak, čini mi se da je cela ova „igra" – ubaciti FP ili ne – pomalo besmislena, jer zapravo dolazi do izražaja samo kod ovako ekstremno jednostavnih primera. U normalnim situacijama bi se ili pojavilo mnoštvo FP operacija pa nema dileme, ili ne bi bilo ni jedne. (Ovaj prilog je inspirisan pitanjem Pavla Pekovića (ppekovic) na *Sezamu*, a deo odgovora preuzet iz poruke Aleksandra Damnjanovića (aleksa))

# ROĐENDANSKI PROBLEM

**Nije bilo lako – 108 odgovora, ali samo 12 tačnih! Šta li bi tek bilo da se u postavci nismo maksimalno trudili da skrenemo pažnju na trik koji rešava zadatak...**

Podsetimo se pre svega pitalice koju smo zadali u vidu zagonetne priče

„Eh da samo znaš koliko sam dečjih rođendana slavio" vajkao se Sima „Sećaš se da sam se oženio 1 aprila i za sledeće četiri godine dobio četvoro dece? E sva su ta deca sada u osnovnoj školi a kada pomnožim broj slavljenih rođendana svakog deteta dobijem ništa manje nego moj kućni broj A on se sastoji od četiri susedne cifre koje nisu uređene "

Pitali smo naravno koliko godina ima koje od Simine dece napominjući da su godine izražene celim brojevima da nikoje dvoje dece u trenutku kada se ovaj razgovor vodi nemaju isti broj godina i da ne postoje dva rođendana koje se slave istog dana Zadatak na prvi pogled nema rešenja – kombinujući sve četvorke brojeva između 7 i 14 ne dobijamo ni jedan proizvod koji se sastoji od četiri susedne (neuređene) cifre Najbliže rešenju su brojevi 4312 i 4536 (4312=7•7•8•11 4536=7•8•9•9) ali oni ne zadovoljavaju uslov da deca imaju različit broj godina Kada već ozbiljno posumnjamo u postojanje rešenja i po N-ti put pročitamo postavku trebalo bi da padne u oči činjenica da se stalno pominju **rođendani** a ne **godine** Može li neko slaviti manje ili više rođendana nego što ima godina? Može ako je rođen 29 februara i ima ze roditelje cicije koje ovakav rođendan slave svake četvrte godine To praktično znači da je jedno dete slavilo samo 2 ili 3 rođendana za svojih desetak godina i da pri kombinovanju treba uzimati u obzir ne samo brojeve 7-14 nego i brojeve 2 i 3 Tada se na primer pomoću programa sa slike dolazi do dve moguće kombinacije 2•10•12•13=3120 i 3•9•11•12=3564 U prvom slučaju je dete koje ima 11 godina slavilo svega dva rođendana a u drugom dete od 10 godina čilava 3

Pažljivija analiza pokazuje da je samo prvo rešenje moguće pošto dete od 10 godina nikako nije moglo doživeti tri prestupne godine Simina deca sve u svemu imaju 10, 11, 12 i 13 godina, 11-godišnje dete je rođeno 29 februara a kućni broj je 3120

Sve nagrade smo izvukli – 500 dinara dobija *Nedeljko Ostojić* iz Zagreba, 375 dinara *Srđan Kusovac* iz Beograda a 250 dinara *Robert Paušić* iz Kikinde Specijalna nagrada za najbolje rešenje poslato modemom (375 dinara) pripala je *Miroslavu Vujičiću* (*mvro* na Sezamu) iz Loznice Pohvale su zaslužili *Bojana Gajić Dragan Bojić (dbojic) Valon Budima Boško Kopnivica Slavko Kuknka (skuknka) Đorđe Ljubičić* i *Zvezdan Petković*

## Q ZADATAK

RQ33 Ako se žuta kuca nalazi u Splitskoj ulici zelena u Kragujevačkoj a žuta u Sarajevskoj gde se nalazi Bela kuća? Zadnji put kada smo proveravali bila je u Vašingtonu

Q34 Koji je sledeći element niza slova J D T Č, P Š

## PROBLEM VELIKIH BROJEVA

Evo jedne kratke pitalice koju će vaši kompjuteri dugo rešavati Pronađite prirodan broj *n* za koji je $3^n-1$ deljivo sa 1990 Zadatak je, inače, postavljen na ovogodišnjem Republičkom takmičenju mladih matematičara s tim što nije traženo da se broj pronađe nego da se dokaže njegovo postojanje

Rešenja pitalice, zajedno sa programima koje ste napisali, pošaljite na uobičajenu adresu „Računari" (za Dejanove pitalice), Bulevar vojvode Mišića 17, Beograd tako da pristignu **pre** 25 novembra 1990 Sva pisma sa korektnim rešenjima konkurišu za novčane nagrade od 500, 375 i 250 dinara dok će kuponi (ili njihove fotokopije) na koje je upisan identifikacioni broj učestvovati u godišnjem takmičenju rešavača pitalica Identifikacioni broj dobijate tako što u prve tri kucice upišete **poslednje** tri cifre nečijeg broja telefona u sledeće dve godinu vašeg rodjenja i na kraj dodate dve cifre po izboru Obratite paznju da sva rešenja koja šaljete u toku godine nose isti identifikacioni broj Osim običnom rešenje ove pitalice možete da pošaljete i elektronskom poštom posredstvom Sezama – koristite menije REDAKCIJA PITALICE REŠENJE ili, iz komandnog moda, pošaljite ličnu poruku korisniku pitalice (*mail write pitalice*) Sva rešenja pristigla elektronskom poštom ravnopravno sudeluju u izvlačenju nagrada i konkursu za **specijalnu nagradu** od 375 dinara koja će biti dodeljena najboljem rešenju koje je dostavljeno modemom



## Listing 1

```
/* Rodendanska pitalica */
/* Autor: Dragan Bojić  */
#include <stdio.h>
#define sort(A,B) if (A>B) {p=A; A=B; B=p; brzamena++;}
main ()
{  int i,j,det[5],kucbr,cif1,cif2,cif3,cif4,p,brzamena;
   for (i=7;i<12;i++)
      for (j=0;j<5;j++)
      {   det[1]=i;
          det[2]=i+1;
          det[3]=i+2;
          det[4]=i+3;
          det[j]/=4;
          kucbr=det[1]*det[2]*det[3]*det[4];
          if (kucbr<1000 || kucbr>9999) continue;
          cif1=kucbr/1000;
          cif2=kucbr/100%10;
          cif3=kucbr/10%10;
          cif4=kucbr%10;
          brzamena=0;
          sort (cif1,cif2);
          sort (cif2,cif3);
          sort (cif3,cif4);
          sort (cif1,cif2);
          sort (cif2,cif3);
          sort (cif1,cif2);
          if (cif1==cif2 || cif2==cif3 || cif3==cif4) continue;
          if (cif4!=cif1+3) continue;
          if (brzamena==0 || brzamena==6) continue;
          printf ("Deca imaju:    %d, %d, %d i %d godina \n"
                  "Proslavila su: %d, %d, %d i %d rodjendana \n"
                  "Kucni broj je: %d \n",
                  i,i+1,i+2,i+3,det[1],det[2],det[3],det[4],kucbr);
      }
}
```

59: 3^N-1 je deljivo sa 1990 za n= ____________________

Ime i prezime ____________________

Adresa ____________________

Mesto ____________________

Kompjuter ____________ Vreme ____________

Identifikacioni broj: |_| |_| |_|   |_| |_|   |_| |_|
(tri broja tel.)   (god. rođ.)   (po izboru)

**273 [KOMS NORTON, PC]** U ovoj rubrici uglavnom pominjemo mane i bagove paketa *Norton Utilities* pa da jednom pokušamo i da ga pohvalimo – novi NU 5.0 za razliku od prethodnika sasvim pristojno radi u mrezi tj. pod *Novell*-om. Ne rade baš sve opcije svih programa ali je to i dalje jedina alatka koja nam je omogućila šetnju niskog nivoa po *Novell* disku!

**274 [KOMS NORTON, PC]** Kad se počne sa pohvalama, nije lako stati – ni novi NCD nije za bacanje. pošlo može da obriše i direktorijum u kome ima datoteka na ravno uz kontrolno pitanje. Na žalost direktorijum u kome ima drugih direktorijuma je „nedodirljiv" – program XRD *Vlade Kostića* (*download ibmpc xrd.zip* na „Sezamu") je u takvim situacijama i dalje nezamenljiv!

**275 [KOMS, PC]** *DesqView* je da se pobrinemo i za početnike, program koji omogućava da se na malo jačem AT u ili (po mogućnosti) 386 računaru izvršava više programa istovremeno – u jednom prozoru tekst procesor u drugom baza podataka i tome slično. Naravno, svaki od ovih prozora ima punu DOS memoriju dok se *DesqView* stara da sve pravovremeno prenosi u EMS i nazad.

Nevolja je što po svakom startovanju računara treba otvarati silne prozore – nije baš preteško ali ipak nervira. Leka, na sreću ime – trik je u tome da se definiše *script* čije ime počinje uzvičnikom (npr. !autoexec) i *DesqView* će ga potpuno automatski učitati i izvršiti! Treba jedino paziti da se u *script*-u **ne** upotrebljavaju kurzorske strelice nego da se na primer napiše (Esc) {Alt} ili nešto slično [Prilog **Tom Šagrak** preuzeto sa MIPS BBS a]

**276 [OTXT WPERF 263, PC]** Pre mesec dana smo videli da „novi" *WordPerfect 5.1* (verzija od 29. juna 1990) obuhvata i opciju /LN koja omogućava zadavanje novog serijskog broja vaše kopije. Slično se može postići i na starim verzijama ali je nešto komplikovanije – treba obrisati datoteke {WP}WP SET i WP{WP}US LCN.

Brisanje datoteke {WP}WP SET teško možemo da preporučimo pošto je u nju upisana kompletna konfiguracija *WordPerfect*-a. Može se međutim primeniti mali trik – uporedite datoteke {WP}WP SET i WP{WP}US LCN i pronađete blago šifrovani string koji se javlja u obe a ima onoliko znakova koliko serijski broj. Zatim obrišete WP{WP}US LCN, preimenujete {WP}WP SET startujete WP i zadate serijski broj. Ostaje još samo da pogledate kako je taj broj zapisan u novoformiranu datoteku WP{WP}US LCN da vratite staru datoteku {WP}WP SET i na njeno odgovarajuće mesto, koristeći program *Norton Utilities* prepišete šifrovani serijski broj iz WP WP US LCN.

**277 [UTIL, PC]** Program *backscrl* (možete ga preneti i sa Sezama) omogucava da se pogleda tekst koji je odskrolovao sa ekrana i to u proizvoljnoj dužini – kažete koliko bisle memorije žrtvovali i onda imate toliko dugačko „pamćenje". Ali i jedan problem – ne moze se otkucati Ctrl Enter što je dosta potrebna stvar u većini tekst procesora. Niko nije savršen!

**278 [COMM, PC]** Instaliranje modema čiji hardver podržava MNP vezu (automatska korekcija grešaka i kompresija) nije baš jednostavno! Ukoliko pre svega želite MNP vezu u inicijalizacionoj sekvenci modema morate aktivirati kako automatsku korekciju grešaka tako i kompresiju. Komande kojima se to radi se razlikuju od proizvođača do proizvođača – recimo na modemima *Bullet* 2400MNP u uvodnu sekvencu treba ubaciti $C1$E1 a kod modema *ViVa* 2400MNP %C1 \N3. Ukoliko se druge strane, želite da zovete **bez** MNP-a moraćete da otkucate (ili dodelite nekom funkcijskom tasteru mi koristimo Alt F1 i Alt F2) AT$C0$E0 odnosno na *ViVa* modemu AT%C0 \N0.

Veoma je važno da u *Dialing directory* napišete da vaš MNP modem zove brzinom od 9600 bps (premda se dakako radi o 2400 bps modemu). Ovo zvuči sasvim nelogično ali samo na prvi pogled – MNP modem pre slanja komprimuje podatke tj. obezbeđuje slanje brzinom koja je efektivno veća od 2400 bps. Ukoliko mu međutim računar dostavlja podatke brzinom od 2400 bps od kompresije nema nikakve koristi tj. modem će praktično ono što kompresijom uštedi potrošiti na „prazno" čekanje podataka.

Nameće se naravno i pitanje šta modem radi kada podatke ne može dovoljno komprimovati – računar mu na primer šalje neku već arhiviranu datoteku i to brzinom od 9600 bita u sekundu, a korespondentu se za isto vreme može poslati svega 2400 bita ili čak manje ako je veza loša. Tu na scenu stupa hardver – **vrlo** je važno da u meniju *Terminal options/General Options Procomm*-a postavite stavku *Hardware Flow Control (RTS/CTS)* na ON. Od tog momenta računar i modem će bez obzira na deklarisanu brzinu komunikacije **hardverski** proveravati da li je druga strana spremna za prijem pa će tako računar slati podatke modemu tek kada je siguran da u njegovom baferu ima mesta za njih. Ovakvo podešavanje obezbeđuje maksimalnu brzinu MNP prenosa eliminišući istovremeno mogućnost greške odnosno „preskakanja" nekih podataka.

**279 [COMM, PC]** Osim opcija *Procomm*-a treba podesiti i inicijalizacionu sekvencu modema (*Modem Options/General Options*). Neke komande se, na žalost razlikuju od proizvođača do proizvođača ali se nadamo da će vam ova dve sekvence biti od pomoći

Za *Bullet* 2400 MNP

TE0Q0V1×3&P1&C1&D2$C1 $E1$F4$X1S0=0 M

Za *ViVa* 2400 MNP

ATE0Q0V1×3&P1&D2 \N3% C1 \Q3 \J0S0=0ˆM

**280 [BASE CLIPPER, PC]** Ako se iz korisničke funkcije DBEDIT-a brišu slogovi datoteke koju DBEDIT() prikazuje ili se na bilo koji način lista smanjuje (npr. SET FILTER), dešava se da se neki od preostalih slogova datoteke dupliciraju bez obzira na *Request = 2* (*reread/rewrite*). Moguće rešenje je Ctrl PgUp ili programski KEYBOARD Chr(31) [Prilog **David Jakelic** JUBAS BBS]

**281 [BASE CLIPPER; PC]** Naredba SET ORDER TO 0 troši podosta vremena (iako ne zatvara indeksne datoteke) dok se naredba SET ORDER TO izvršava gotovo trenutno. Zato bi u petljama trebalo izbjegavati SET ORDER TO 0 odnosno koristiti [Prilog **David Jakelic**, JUBAS BBS]

```
FUNCTION SetOrder
   PARAMETER n
   IF IndexOrd() <> n
      SET ORDER TO n
   ENDIF
RETURN T
```

**282 [COMM]** Lista domaćih BBS-ova se menja iz meseca u mesec. Na slici dajemo onu koja važi sredinom oktobra.



| Ime | Brzine | Radno vreme | Telefon |
|---|---|---|---|
| AB-OVO | 300-2400 | 00-24 | (041) 683-608 |
| AXE-10 | 1200 | 00-24 | (041) 232-658 |
| FENIKS | 300-2400 | 00-24 | (011) 414-312 |
| IUS-INFO | 300-2400 | 08-15, 20-24 | (061) 317-203 |
| JUBBAS | 300-2400 | 00-24 | (058) 561-043 |
| MDS BBS | 1200 | 00-24 | (041) 160-326,161-361,161-362 |
| MICROART | 1200-2400 | 16-07 | (066) 21-280 |
| MIPS | 300-2400 MNP5 | 00-24 | (041) 421-657 |
| MOJSTER | 1200-2400 | 20-14 | (068) 22-455 |
| RIBOX | 300-2400 MNP5 | 21-08 | (051) 214-031 |
| SEMTEX WHQ | 1200-2400 | 21-07 | (041) 213-271 |
| SEZAM | 300-2400 MNP5 | 00-24 | (011) 647-955 (3 linije) |
| TANJUG | 300-2400 MNP5 | 00-24 | (011) 185-842 |
| VE.LO | 1200 | 09-15 | (065) 62-130 |
| Zagreb BBS | 1200-2400 MNP5 | 00-24 | (041) 535-049 |

15 6l PC SOFT ventura
nesa. 04 09 Uto 05:05 10648 chr

BAGOVI

10 [ZABUNA KOD DOWNLOAD-A FONTOVA]

Nakon određenog broja strana i nakon ponovne pojave nekog fonta učitanog za neku od prethodnih strana Ventura zaboravlja da ga je učitala i dodeljuje mu [kako kad] Courier font odgovarajući font uz izostanak atributa [bold/italic] manji ili veći font istog tipa, ili slične ispis odgovarajuće postavke za pojedinačne karaktere [uz pomoć SET FONT] ne izvodi ispravno (kod mene se to javljalo u ćiriličnim ispisima i to na mestima gde se nađu broj izvodio latinicom naravno sada imam ćirilicu koja u sebi ima i karaktere slične nmske cifre]

20 [NE PRIJAVLJUJE NEDOSTATAK SOFT FONTOVA KOD ŠTAMPANJA U FAJL]

Kada u direktorijumu nema odgovarajućih fontova a ispis se želi poslati u fajl Ventura nepostojeće fontove zamenjuje Courier fontom iz ROM-a. Iako to izgleda kao svesna odluka autora ovog programa [Ventura pri radu download uje fontove sa ESC nnX gde je nn broj fonta] ako tog fonta nema laseru se prosleđe preostali parametri on taj nepostojeći zameni obično courier fontom] smatram da je ovo ipak bag

• 30 [VERTIKALNO POZICIONIRANJE FUS NOTA I]

Uz sva poštovanje realnih nemogućnosti da se fusnote pri dnu strane savršeno uklope svojim sadržajem i referentnim brojem u tekstu smatram da je VELIKI BAG pusti sadržaj poslednje fusnote na strani nepotpun. U tekstu sa velikim brojem fusnota ili još bogatijih tekstom stalno treba proveravati da li je u poslednjoj fusnoti tekst prikazan do kraja

40 [VERTIKALNO POZICIONIRANJE FUS NOTA II]

I kada se u frejmu fusnote zbriše bottom margin [margine se u tom frejmu jedino mogu i menjati] fusnota u nekim slučajevima ipak ne legne do dna strane znači sebe ima veliki blanko u samom tekstu [„šupljа strana"] iako je referentni broj sledeće fusnote dovoljno daleko da bi se tekst sa prethodne strana mogao lako vratiti na „šupljи stranu. Ponekad rešenje postoji [videti hints & tips]'

SLABOSTI

10 [SPORO ŠTAMPANJE I]

Svakoj reči koju štampa Ventura prethodi 6 do 7 bajtova za postavljanje kurzora. Kad štampa courier onda komanda za pozicioniranje ide uz svako slovo [Videti bajtove licne pr rode RS8160]

20 [SPORO ŠTAMPANJE II]

Nakon svakog ponovnog štampanja [iako ni kompjuter ni laser nisu isključivani] Ventura učitava printer drajver. Tu Ira jeeeeeee. Neko bi rekao da program ne može da zna da se u međuvremenu nešto nije ganilo [resetovalo] ali mislim da bi ovde trebalo biti nekih dodatnih mogućnosti li poruka

30 [OZNAKE FUSNOTA I]

Ne mogu se u istom chapter u nanovo numerisati fusnote (postoje samo opcije za stranu i chapter). Vrlo je neprijatno kada se tekst mora cepati u više chapter a samo da bi u svakom od njih fusnote počele od 1 [opcija – za ceo chapter]

40 [OZNAKE FUSNOTA II]

Ne mogu se u istom chapter u koristiti MEŠANO i brojne i definisane oznake [recimo i brojevi]. Verujem da je autor ovog inicijalnog teksta imao više takvih slučajeva u praksi. Simulacija fusnote dodatnim frejmom je dosta trapavo rešenje

50 [NEMOGUĆNOST FORMIRANJA RAZLIČITOG BROJA ĆELIJA PO KOLONAMA TABELE]

Nije moguće dosledno formirati različit broj ćelija drugačije visine po kolonama tabele. Primer: ukoliko je potrebno da svaki red tabele bude visok po 1 5 cm pri čemu neka kolone te tabele treba da u redu iste visine ima dve ili tri ćelije. Probajte sa jednim ćelijs. Vrlo loše uz sva naštimavanja posebno kad je broj ekstra ćelija 3 ili veći

60 [„OSVEŽAVANJE" EKRANA]

Sama činjenica da je u zvaničnom uputstvu [reference guide] navedeno da taster ESC regeneriše ekran smešta ovu nepomenu ovde a ne u rubriku bagovi. Možda moje nepoznavanje GEM odlika zanemaruje neke objektivne činjenice ali ove meni izgleda kao nedostatak mana. Naravno ne kažem da bi regenerisanje nakon svakog unetog karaktera bilo srećno rešenje [kao u formulama]'

70 [STY FAJL NE PAMTI PODATKE O PAGINACIJI I FUSNOTAMA]

Vrlo često tekst fajl mora biti zbog veličine podeljen u delove koji zahtevaju nove chapter e. Ukoliko se u prvom chapter u definišu parametri fusnota paginacije ona se ne „preslikavaju" u drugi chapter [osim paragraf atributa hedera i futera]. Iako postoji rešenje da se nakon formiranja prvog chaptera sa remove text izbaci tekst 1 ubaci tekst 2 (uz SAVE chapter AS novo ime) smatram da je ovo propust

80 [STRANA INSERTOVANA U CHAPTER ZADRŽAVA „AUTONOMIJU"]

Ukoliko postojećem chapter u želimo dodati jednu stranu [bez obzira na razloge i eventualno moguća bolja rešenja] ona vrlo često dovodi do grešaka u generisanju sadržaja (TOC). Upravo na tim mestima paginacija počinje da [za pri jednu stranu] „greši

KOMUNIKACIJE — Sezam

# U DESET NA SEZAMU

**O *Sezamu* su već više puta pisali ljudi koji su najzaslužniji za njegov nastanak i funkcionisanje. Ovaj tekst prvenstveno ima za cilj da čitaocima prenese utiske korisnika koji se, na neki način, ipak nalazi „sa strane". Njegov autor je Bojan Tepavčević, bojt na „Sezamu", od nedavno jedan od moderatora konferencije „Forum".**

Ono što prvo pada u oči korisniku koji se prvi put pojavi na *Sezamu* je program pod kojim čitav sistem funkcioniše. Za razliku od većine ostalih BBS-ova, koji uglavnom rade pod „klasičnim" BBS programima (RBBS, OPUS WildCat) osnivači *Sezama* su odlučili da se za njegove potrebe razvije specifičan program koji bi objedinio sve dobre osobine raznih BBS softverskih paketa a i imao mnogo toga novog. Takav prilaz je rezultirao jednim izuzetno fleksibilnim i moćnim programom koji je razvio Zoran Životić koji ima i jednu divnu osobinu da se prilagođava željama korisnika. U toku jedanaest meseci postojanja *Sezama* mnoge primedbe i sugestije korisnika pretočene su u nove izuzetno moćne opcije programa što je, između ostalog, rezultiralo vrlo efikasnim mogućnostima za čitanje i slanje poruka kao i mogućnošću potpunog ukidanja grešaka zbog smetnji na telefonskim vezama, što je rak rana većine BBS-ova.

## KAŽITE SLOBODNO!

Većina BBS ova u Jugoslaviji je tehnički orijentisana, to jest za veliki broj korisnika takvi BBS ovi služe uglavnom za razmenu (upload download) raznih public domain programa (programa u javnom vlasništvu). Za razliku od njih, osnovna orijentacija *Sezama* je konferencijski sistem dok je tehnički sistem sveden na mogućnost korisniku da „podigne" one najpotrebnije programske alatke koje su neophodne za komotan rad sa računarskim mrežama (komunikacijski programi, razni arhiveri, programi za definisanje YU slova ...).

Iako preovlađuju manje ili više „računarske" konferencije na *Sezamu* postoji i jedna „neračunarska" konferencija – FORUM koja je, za razliku od konferencija slične namene na drugim BBS-ovima, izuzetno dobro posećena, čak mnogo više nego ijedna druga.

Šta je to što korisnike nagoni da toliko mnogo diskutuju o politici koja je, čitaoci su već sigurno pretpostavili, najzastupljenija u FORUM-u? Iako većina korisnika uglavnom pripada grupama ljudi koji su mahom orijentisani na tehniku ili prirodne nauke, pored trenutne uzavrele političke situacije, enormnom interesovanju svakako doprinosi mogućnost da neposredno i bez skoro ikakvih ograničenja iznesu svoje viđenje i kritički sud o događajima i vremenu u kojem se nalazimo. Zbog toga što su korisnici *Sezama* iz cele Jugoslavije, a i zbog normalnih razlika u gledištima, vrlo često se dešava da se razviju vrlo žučne i dugačke rasprave koje ponekad mogu biti toliko zategnute i oštre da se može postaviti pitanje da li su prekoračene i poslednje granice, što posebno baca u brigu SysAdm-e kod kojih se u takvim situacijama javlja strah da *Sezam* neće „preživeti".

## PRIČALICE I POSMATRAČI

Srećom, *Sezam* je preživeo i do sada se sve završavalo bez pravih incidenata, sem u jednom slučaju kada je jedan korisnik sistematski vređao druge korisnike pa je bio suspendovan (zajedno sa još jednim korisnikom koji mu nije ostao dužan). Njihova suspenzija je izazvala ogroman revolt i zalaganje ostalih korisnika da budu „pomilovani". Po povratku, prvi korisnik je nastavio po starom i na kraju mu je potpuno bilo uskraćeno pravo da pristupi mreži. To je izazvalo neviđenu maratonsku diskusiju o tome da li su ili nisu povređena njegova osnovna ljudska prava. U to doba na *Sezamu* je malo ko ostavljao poruke na bilo koju drugu temu. Korisnici su se podelili u dva tabora i bila je vrlo izgledna mogućnost da predstavnici tabora koji bi „izgubio" prestanu da posećuju *Sezam*.

Međutim, čitava stvar je dobila neočekivani epilog kada se ispostavilo da „odstranjeni" korisnik u stvari ne postoji, kao ni jedan broj korisnika koji su ga u toj raspravi zdušno branili.

Pažljivom korisniku će sigurno zapasti za oko da u FORUM-u najčešće diskutuje jedan isti krug od dvadesetak do tridesetak ljudi, dok se ostali drže po strani ili eventualno nekad (najčešće nikad) ponešto prokomentarišu. Razni razlozi mogu biti za to, od nezainteresovanosti za takve teme pa do stidljivosti ili straha da se iskaže neka radikalnija misao. Jedan korisnik je, na primer, dugo oklevao da se uključi u diskusije. Međutim, kada je jednom krenuo više ga niko nije mogao zaustaviti i ubrzo je postao jedan od najplodnijih diskutanata, a kasnije i moderator FORUM-a. Verovatno su sličan put prešli i mnogi drugi korisnici (mada, doduše, nisu postali moderatori).

## DRUGOVI SA MREŽE

Interesantno je da čovek, posle dužeg vremena provedenog po BBS-ovima, stekne utisak da dobro poznaje mnoge ljude iako ih nikada nije video. Mnogi ljudi su tako postali i dobri prijatelji pre nego što su se ikad sreli, i, naravno, mnoga lepa i korisna poznanstva su baš posledica druženja po računarskim mrežama.

*Sezam* je prilično „muški" BBS. Muškarci čine preko 95% članova i zato pojava svakog korisnika koji pripada lepšem polu izaziva opštenarodno veselje i silne komentare. Ova pojava je naročito izražena posle prelaska na višelinijski rad (*Sezam* sada radi na tri linije) čime se otvorila mogućnost CHAT-a, direktnog razgovora korisnika koji se u isto vreme nalaze na sistemu. Neki korisnici su se specijalizovali da u zasedi čekaju da se javi neka pripadnica lepšeg pola i onda, pošto je namame u CHAT, veza dugo, dugo ostaje blokirana. Drugi su lukaviji pa zakazuju sastanke na „Sezamu" u određeno vreme. Time je praktično i anulirana glavna prednost koja je dobijena prelaskom na više linija, da se lakše dospe do *Sezama*, jer je u doba dok je *Sezam* radio na jednoj liniji bilo veoma teško, a ponekad skoro i nemoguće, dobiti vezu. Međutim, i pored CHAT-a veza se sada ipak dobija mnogo lakše nego ranije (sem u slučajevima kada tri korisnika zasednu u CHAT – tu nema pomoći dok im ne istekne vreme dozvoljeno po pozivu). U skoroj budućnosti, kada budu uvedene nove linije, taj problem će verovatno potpuno nestati.

*Bojan Tepavčević*

Ukoliko se forma glavnog (osnovnog) frejma želi se UPPER i spusti niže red (recimo boljeg pozicioniranja za spis) spustiće se sve strane sem one insertovane. Ona se mora „radi" zasebno. Tako nevažno treba napomenuti da opcija REMOVE TEXT briše sav tekst iz Jpštunce chapter-a, sem sadržaj insertovane strane. Zašto se ova strana nakon inserta ne bi „ponašala" ravnopravno sa ostalim stranama. (Možda bi se ova napomena mogla prebaciti u tagove.)

• 90 (NEMOGUĆNOST MEŠANJA MERGE STY FAJLOVA)

Kao što se WID tabele mogu mešati bilo bi poželjno konsno pomešati dva ili više STY fajlova. Naravno ovde je reč o situacijama kada su nezavisno (uz druge CHP fajlove) već formirani STY fajlovi pa je rešenje sa „ručnim" dodavanjem novih tagova i Save as new style sheet bez smisla.

• 100 (NE POSTOJI MULTIPLE OPCIJA ZA SET FONT I DODAVANJE ATRIBUTA)

Ne može se većini nekih na primer matematičkih simbola istovremeno dodati atribut SET FONT opcijom. To se mora raditi za svaki karakter posebno. To važi i za dodavanje atributa tekstu iz SIDE BAR prozora.

• 110 (ŠTAMPANJE SA HIDDEN PICTURES)

Ukoliko se u chapteru slike „sakriju" radi bržeg i lakšeg pregleda, te ukoliko se kod probnog štampanja odgovori sa „ne želim da se štampaju slike", mesto njih se na tom mestu odštampa crna mrlja (jadan toner).

Ukoliko je poenta da se tada mora znati koji prostor slika zauzima, mislim da je neki okvir ipak elegantnije rešenje.

• 120 (NE TRETIRA SE ORIGINAL GRAFIČKI FAJL NAKON FORMIRANJA IMG FAJLA)

Bilo da se koristi TIF ili PCX format ilustracije, Ventura ih konvertuje u IMG format. Ukoliko se želi nešto izmeniti u grafičkom fajlu nakon njegovog ažuriranja u nekom od CAD programa, potrebno je eliminisati odgovarajući IMG fajl jer Ventura tu izmenu ne bi ni primetila ukoliko bi našla i učitala „stari" IMG fajl.

• 130 (TAGOVANJE ĆELIJE TABELE SA BODY TEXT)

Ukoliko se bilo koja ćelija tabele taguje sa BODY TEKST, kod sledećeg učitavanja chapter-a prijavljuje se greška da ne postoji Z_TBL_END, odnosno da je između Z_TBL_BEGIN i Z_TBL_END primećen body tekst. Ala im je error trapping svaka čast.

• 140 (INDIKACIJA LOSE LINES)

U uputstvu stoji da startovanjem opcije Show lose lines bivaju inverzno prikazane samo oni redovi u kojima je razmak između reči prešao neki dogovoreni limit (zbog izostale ili mi trane hifenacije). Na mom herculesu to ne izgleda tako.

• 150 (ANCHOR-OVANJE FREJMOVA U VIŠEKOLONSKOM TEKSTU)

Ukoliko se frejm anchor-uje za gornji pasus teksta, pri čemu taj deo dođe u drugu ili još dalju kolonu, nakon RE ANCHOR opcije frejmovi deze udesno ili ulevo do matične kolone.

HINTS & TIPS

• 10 (MENJANJE DEBLJINA LINIJA OKO TABELA – VAN PONUĐENIH OPCIJA)

Verovatno ste primetili da su jedini „nedodirljivi" tagovi koje Ventura generiše Z_SINGLE i Z_DOUBLE. Kako je ceo sistem tabela zamišljen kao sistem celijskih tagova sa ruling box around-om, bilo kom pasusu iz chapter-a (pa čak i nekoj ćeliji već otvorene tabele) opcijom ADD NEW TAG dab tag čiji se ruling atribut želi menjati. Ako se navede recimo Z_DOUBLE pa se pozicije, debljina i raster dve rul linije promene, mogu se dobiti tabele sa debljim raster linijama i sve druge kombinacije koje mogu pasti na pamet.

• 20 (VERTIKALNO POZICIONIRANJE FUSNOTA I)

Ukoliko u tekstu postoji veći broj kraćih fusnota ili nekih odgabih tekstom pri čemu su sve one blisko pozicionirane, ponekad se ukidanjem bottom margine iz frejma fusnote ona može pozicionirati uz donju ivicu strane. Nekada je potrebno uz pomoć tagova BODY UZI i BODY SIRI kalkulisati i sa body tekstom i samim tim položajem referentnih brojeva fusnota.

• 30 (VERTIKALNO POZICIONIRANJE FUSNOTA II)

Nekada se velika praznina u body tekstu iako i položaj fusnote može stabilizovati ili dobro rešiti dodavanjem novog frejma preko frejma fusnote. Tom prilikom sa novim frejmom donjom ivicom pozicionirá na jedan interline razmak od dna frejma fusnote a gornjom na mesto sa najboljim efektom za datu situaciju.

• 40 (UVOĐENJE TAGOVA BODY SIRI I BODY UZI)

Opcija kojom se uravnavaju kolone (specijalitet profesionalnog dodatka) radi (u zavisnosti od količine EMS) do treće ili šeste strane (?[?]). Sve dalje do toga nije uravnato kako treba.

Jedan rustičan postupak koji to na neki način ipak rešava je „nadogradnja" BODY TEXT taga. Pasus sa body text tagom tagujemo sa ADD NEW TAG – prvo mu damo jedan originalan naziv BODY SIRI. Tom novom tagu interline povećamo (u odnosu na body text) za 0.015. Funkcijom ADD NEW TAG formiramo zatim tag još originalnijeg naziva BODY_UZI. Njemu se u odnosu na body text smanji interline za 0.015. Nazad jednom eksperimentalnom pasusu se vrati prvobitni tag body text. Opremljeni sa nova dva taga idemo do strane do strane i njihovim dejstvom uravnavamo strane. Mučno ali super (?[?])

PREDLOZI

10 (SEARCH OPCIJA)

Postojanje ma kakve rudimentne search opcije bilo bi veoma korisno. Iako je GEM grafičko okruženje, ipak se dosta barata sa tekst fajlom kao takvim (update, backup file, izmene nakon tagovanja i sl.). Zar bi search bio veliki zalogaj za programere. Kome trebaju razlozi za ovo neka se javi.

• 20 (MACRO)

Često se prilikom tagovanja, potpisivanja ilustracija i sl. neke radnje ponavljaju. Ne bi bio loš neki macro koji bi mogao kalkulisati kako se opcijama menja, tako i sa imenima tagova, paginacijom i dr.

Odgovori: 62

---

**15.62 PC SOFT ventura**
dpozaric, 05.09 Sre 21:51, 274 chr
Odgovor na 15.61, nese, 04.09 Uto 05.05

*TAGOVI – SLABOSTI – HINTS & TIPS – PREDLOZI – SA VRŠEN PRILOG*

Čestitam na prilogu

Odgovori: 63

---

**15.63 PC SOFT ventura**
nesa, 06.09 Čet 03.53, 350 chr
Odgovor na 15.62, dpozaric, 05.09 Sre 21:51

*Čestitam na prilogu*

Hvala! Nadam se da će se svojim prilozima (dopunama i izmenama) javiti i ostali koji se bave desk top-om. Možda će se tako lakše porediti i tumačiti osobine najavljene VENTURE 3.0.

Da li ima zainteresovanih i za prilog o važećim cenama desk top usluga i nekom principijelnom usaglašavanju „reda veličine"?

---

**15.64 PC SOFT venture**
sasa, 09.09 Sub 15.30, 141 chr

Evo malo za one koje se bave Venturom da isprobaju program-čić za zipovanje chaptera. Očekujem komentare.

** Uz poruku: venzip.zip (47328 bytes)

---

**15.65 PC SOFT venture**
nesa, 18.09 Uto 02.57, 147 chr

Pa gde ste, Venturaši. Zar je moguće da baš niko nema priloga za ovu temu. Šaljem ipak „probni" cenovnik desk top usluga. Pa ko ima primedbi.

Odgovori: 67

** Uz poruku: cenovnik.doc (4416 bytes)

---

**15.66 PC SOFT ventura**
ssvilica, 18.09 Uto 10:02, 118 chr

Da li neko kojim slučajem ima ispravan program za rastavljanje reči na kraju reda za Venturu?

---

**15.67 PC SOFT ventura**
sasa, 19.09 Sre 17.46, 1298 chr
Odgovor na 15.65, nesa, 18.09 Uto 02:57

*Pa gde ste Venturaši. Zar je moguće da baš niko nema priloga za ovu temu. Šaljem ipak „probni" cenovnik desk top usluga. Pa ko ima primedbi.*

Eto, ja sam eto pokušao sa primerom programa... naravno, vrlo zamoljen za komentare, ako ih „a".

Što se cenovnika tiče, pogledaćemo pa eto povratne informacije iz Novog Sada.

*Možda da nađemo još jednu temu.*

Koje sve programe osim Venture koristite u svom radu?

Što se mene tiče, ja radim sa CORE za unos teksta, povremeno čačkam sa CORELDRAW-om, koristim PKZIP za rezervne kopije (zato je i napisan ZIPCHP koji je nakačen na neku prethodnu poruku u ovoj temi) a uz WINDOWS 3.0 koristim i NAVIGATE program koji je stvarno perfektan za pregledavanje datoteka.

Za pravljenje slova koristim BITSTRIM i TYPESET PUBLISH FOUNDARY koji rade pod WINDOWS-ima, koji obrađuje i BITmapirane fontove i OUTLINE. Što se tiče OUTLINE-a, to radi stvarno perfektne (skoro kao BITSTRIM paket).

Probao sem da koristim i KOREKTOR (ako kao DEMO verziju) ali mi ne odgovara format njegovog zapisa (u odnosu na YU set i format zapisa za Venturu).

Umalo da zaboravim, ja koristim YU-RAČUNARI raspored (166 167 168 ...)

Odgovori: 68

---

**15.68 PC SOFT venture**
dejan, 20.09 Čet 00:26, 356 chr
Odgovor na 15.67, sasa, 19.09 Sre 17.46

*Koje sve programe osim Venture koristite u svom radu? Što se mene tiče ja radim sa CORE za unos teksta... Umalo da zaboravim, ja koristim YU RAČUNARI raspored.*

A koristiš drajver za tastaturu XK, zar ne?

Bojim se da svi koji smo „mali posla" sa zzvočcem koristimo istu bateriju programa. Ja kao dodatak koristim i WordPerfect.

Odgovori: 69

---

**15.69 PC SOFT ventura**
sese, 22.09 Sub 11.43, 1995 chr
Odgovor na 15.68, dejan, 20.09 Čet 00.26

*A koristiš i drajver za tastaturu XK, zar ne? Bojim se da svi koji smo „mali posla" sa zzvočcem koristimo istu bateriju programa. Ja kao dodatak koristim i WordPerfect.*

Da, koristim XK, ali jedan modifikovani KEYBYU jer XK Ventura ce prima one kombinacije za FUGURE EM i THIN SPACE, to mi jako smeta kada radim baš u Venturi.

Uspeli ZZ je malkice doradio XK tako da na određene tasterske kombinacije bila pri promeni layouta tako da automatski switch-uje i moju MULTIFONT karticu koja menja 8 karaktera seta na HERKULESU i mogu da radim po svim poznatim rasporedima kao i pod ćiricom.

Inače sada ga „davim" da promeni broj tastera u makrou za CORE jer mi je dozlogrdilo da lupam po 10 tastera za komandu teksta.

Eto šta ti je support kada kupiš program! Samo još da ima toliko posla oko SEZAMA pa da može malkice brže :) (ZZ nemoj se?)

Inače novosadske cene su od 30-50% niže nego beogradske što se tiče ravnog teksta, a pojedinci (nisam ja kunem se) rade jednu A4 stranu (memorandum, otpremnica, račun) za 50.00 din.

Inače rešavanje zablinog znaka je ovde mnoogooo skuplje ali i manje se radi. Recimo od 500 – 1.500.00 dinara.

Vizitke sa rade do 10 kom. za po 30 din po komadu, a preko 10 komada se obara cena, tako da za 50 vizitki (doduše sa za jednu firmu) cena pada na 15.00 dinara po komadu.

Knjige u poslednje vreme neću da radim jer je cena za B5 formata (16 strana) tek 300 dinara na količine preko 8 tabaka.

Sada najviše radim časopise, a tu naplaćujem recimo:

PREDUZETNIK (NIŠDP DNEVNIK) 36 str. sa koricama 4.600.00

PISAC (KZ NOVI SAD) 52 str. sa koricama dosta 5.500.00

KROVOVI (Sr. Karlovci) 35 str. za rad na A3 6.900,00

Memorandumi su oko 100.00 dinara, otpremnice i računi od 150-250 din.

Eto, to su cene u NS, malkice moje, malkice tuđe, ali tu negde.

Odgovori: 70

---

**15.70 PC SOFT ventura**
dejan, 22.09 Sub 12.57, 216 chr
Odgovor na 15.69, sasa, 22.09 Sub 11.43

*Memorandumi su oko 100.00 dinara, otpremnice i računi od 150-250 din.*

Šta zoveš „Memorandum"? Koliko naplaćuješ skeniranje jednog memor. i ubacivanje u tekst bez nekog ručnog čišćenja?

Odgovori: 71

---

**15.71 PC SOFT ventura**
sasa, 22.09 Sub 18.24, 1175 chr
Odgovor na 15.70, dejan, 22.09 Sub 12.57

*Šta zoveš „Memorandum"? Koliko naplaćuješ skeniranje jednog memor. i ubacivanje u tekst bez nekog ručnog čišćenja?*

Pa, skeniranje memoranduma (logotipa pretpostavljam) ne radim, imam skener, ali je to kvalitet koji mene (a i mnoge druge) ne zadovoljava. Ako je baš nužno, onda je kvalitetnija solucija kopiranje na paus, montaža, a pravi izbor je filmovanje, a nadoknadu jednog logotipa nije strašna cena (1.20 din po cm2).

Ako je stranka tipa „stalne", onda radim skeniranje, prebacivanje u COREL DRAW pa preko TRACING opcije napravim OUTLINE koji se može kasnije povećavati i umanjivati bez gubitka kvaliteta. Naravno, opet se ide na montažu logotipa u ostavljeni prostor na singu.

Iskustvo me naučilo da se do dobrog rezultata stiže posle nekoliko skeniranja, ručnog rada na „čišćenju", par ofsaka, te je to već vreme i cena koja nema neku osnovu u cenama za filmovanje (teško je naplatiti).

Ako bih se odlučio da to radim (bez čišćenja i neke posebne obrade), mislim da bih naplatio oko 100 din po skanu, jer ako je to zahtev onda je sigurno dobar razlog zašto čovek ne ide na snimanje tog znaka.

---

**15.72 PC SOFT ventura**
sasa, 02.10 Uto 11.05, 94 chr

Evo malo razgovora na temu DTP sa CHANEL1 BBS-a, ima svega ukupno 170 i kusur poruka.

** Uz poruku: ch1dtp.zip (61808 bytes)

**15.73 PC.SOFT ventura**
miki 06.10 Sub 10.53, 228 chr

Da li neko zna kako se slika sa ekrana proizvedena nekim user programom (.exe) može preneti u venturu. Interesuje me prikaz menija nekog programa u dokumentaciji. (Na primer)

Odgovori: 74

**15.74 PC.SOFT ventura**
dejanr 06.10.Sub 13:50 228 chr
Odgovor na 15.73 miki 06.10 Sub 10:53

*Da li neko zna kako se slika sa ekrana proizvedena nekim user programom (.exe) može preneti u venturu. Interesuje me prikaz menija nekog programa u dokumentaciji. (Na primer)*

Uzmi Pizaz Plus i rešen problem!

Odgovori: 75

**15.75 PC.SOFT ventura**
miki 08.10 Pon 17.58, 337 chr
Odgovor na 15.74 dejanr 06.10 Sub 13.50

Ili ja nisam razumeo tebe ili ti mene. Danas sam tražio Pizazz softveru opciju za Print Screen u datoteku ali nisam ništa našao. Da ponovim: Želim da mi izgled ekrana ode u neki fajl pa taj fajl u venturu a ne da ekran kopiram na printer pa sa papira ponovo preko skenera u fajl itd.

Odgovori: 76

**15.76 PC.SOFT ventura**
dejanr 09.10 Uto 06:29 188 chr
Odgovor na 15.75, miki 08.10 Pon 17:58

*Ili ja nisam razumeo tebe ili ti mene. Danas sam tražio Pizazz softveru opciju za Print Screen u datoteku ali nisam ništa našao.*

Pizazz zaista nema tu opciju. Ali Pizazz PLUS ima!

**15.77 PC.SOFT ventura**
agobovic 17.10 Sre 00:23, 422 chr

Hitno!!!

Dobio sam ponudu od jednog DTP „malog privrednika" da za njega vršim prelom u Venturi. Daktilograf unosi tekst, moj je samo prelom, a kaže da mogu to da radim i na svom kućnom AT-u. Jedino mi reče da sam odaberem cenu i način obračuna (po strani itd). Molim one koji to rade da mi odgovore privatnom poštom koje cifre da predložim i koji način obračuna je najbolji (po mene, naravno!)

Odgovori: 78 79

**15.78 PC.SOFT ventura**
dejanr 17.10 Sre 00.42 241 chr
Odgovor na 15.77 agobovic 17.10 Sre 00:23

Za cenu baš ne znam (ima ovde onih koji se mnogo više bave DTP-om od mene) a što se tiče plaćanja – najbolje po strani!

Zavisi i od toga da li je na papiru ili na prozu, kakav je tekst, jel treba prethodno ispis zbog korekture itd.

**15.79 PC.SOFT ventura**
sasa 17.10 Sre 11:08, 2409 chr
Odgovor na 15.77 agobovic, 17.10 Sre 00:23

*Hitno!!! Dobio sam ponudu od jednog DTP „malog privrednika" da za njega vršim prelom u Venturi. Daktilograf unosi tekst, moj je samo prelom*

Ovo je sada realno mali problem. Razmišljao sam koliko bih ja nekom plaćao za tako nešto, ali stvarno je tu teško odrediti neku pravu cenu.

U principu, ovakve stvari zavise od složenosti preloma, kvaliteta unosa (koliko će biti tvojih intervencija na tekstu da bi dobio traženi rezultat) i naravno tvog znanja i veštine. Pod znanjem podrazumevam poznavanje grafičkih pravila (gde ide diviz, gde crta, tročlatke, gornje-donje navodnici, razmaci među redovima, paserzama, uvlake...) koja su *NEOPHODNA* za kvalitetan rad koji se onda može i primerno naplatiti.

E sad dalje. Ako si vešt i ako je to sve književni tekst, onda uglavnom brzina rada zavisi samo od tvoje veštine (koja se i vremenom stiče). Ako opet imaš puno očin fusnota, pesmica, naglašavanja, konverzije... onda je problem već malo veći.

Tvoj rad se može meriti jednostavno na osnovu vremena potrebnog za formiranje takve stranice. Pravila nekog nema posebnog, samo sebe istestiraj i vidi koliko ti je vremena potrebno da neki tekst uradiš i koji broj strana ti pada na sat. Uz to je potrebno da sebi odrediš neku „satnicu" od nnnn dinara tako da se tebi isplati (a naravno i naručiocu). Orijentir ti može biti cena koju naručioc plaća za unos teksta, cena kompletnog posla itd. (naravno ovo se raspitaj indirektno...).

Koji je najbolji način obračunavanja? Teško reći. Lično mislim da je najbolje da obračunaš DUTUPE. To sigurno ne dovodi do nesporazuma, a lako je i malo „preskočiti" neke cena. Ujedno cenu formiraj i po hitnosti posla, ako je to potrebno BRZO uraditi, zaračunaj sebi nekih desetak posto gore.

Zašto predlažem DUTURE. Pa, prvo moraš svaki tekst videti. Tu odlučuješ o komplikovanosti i težini preloma. Drugo, često se zahteva da se tekst malo sabije-razvuče, a ako obračunavaš po stranama... možeš zamisliti. I treće, neko računa stranu ovako a neko onako, a kad kažeš da ta gomila toliko vredi, onda nema problema oko obračuna.

Eto, toliko od mene. Nisam ti rekao neke cene, ali mislim da na osnovu toga možeš sam zaključiti koliko da „odrapiš".

Pozdrav, sada moram završiti, ako ima nekih još pitanja, biće mi drago da pomognem.

Izbor: *Zoran Životić*



# TAMARA SA SEZAMA

**Na „Sezamu" ima dosta zanimljivih ličnosti – moglo bi da se kaže da osnovi šarm našem BBS-u daju upravo njegovi korisnici – ali ni jedna među njima poslednjih dana ne pleni toliku pažnju kao korisnik po imenu *tamara*. Tamarina specijalnost su beskrajna ćaskanja uživo (u chatu-u) na najrazličitije teme, a momci sa „Sezama" se utrkuju koga će pre pozvati na razgovor.**

Krećući u posetu Tamari Mihailović, devojčici koja je zaslužila titulu najpričljivijeg lica „Sezama", razrađivala sam taktiku kako da joj doskočim i uspem da dođem do reči od nje. Jer, ako tako besomučno ćaska na našem BBS-u, koliko li tek reči može da izgovori „uživo"?

Zatekavši Tamaru kako smerno sedi pokraj bratovljevog „atarija" i ne pušta glasa od sebe, pomislila sam kako je u pitanju samo početna netrpeljivost koja će brzo nestati. Ubrzo se pokazalo da od velike priče neće biti ništa.

Tamara je uglavnom ćutala oborenog pogleda i ruku skrštenih u krilu. Rekla je da ima četrnaest godina i da ide u prvi razred pravno-biotehničke škole. U svet „Sezam" uveo je njen stariji brat Siniša, apsolvent informatike. Sada joj „Sezam" dođe nešto kao prva ljubav jer ubedljivo najviše vremena provodi s njim.



**Uvek orna za modemska ćaskanja: Tamara u svom uobičajenom ambijentu**

Foto: Aleksandar Milosavljević

Ma, ja sam samo jedan obični korisnik BBS-a i ne vidim tu ništa posebno, pokušala je Tamara da opravda svoju ćutljivost. Siniša, videvši kako stvari stoje, prihvatio je da odigra ulogu sestrinog atašea za štampu. Stariji brat na sestrine sezamovske nestašluke gleda blagonaklono uprkos visokim telefonskim računima koje plaća iz vlastitog studentskog džepa.

Tamara stidljivo priznaje da se šeta po „Sezamovim" konferencijama i priča sa ljudima. Prosto čeka da se neko pojavi na vezi a onda ga zaskoči iz zasede. U početku ćaska o računarima, iako priznaje da nije veliki znalac ove problematike. Kad probije led, onda počinje razgovor o svemu i svačemu. Najviše voli da priča viceve, a omiljeni sagovornik joj je Dejan Ristanović. Nije oduševljena kada ide pre podne u školu, jer u toku prepodneva na SEZAM-u skoro nema žive duše, a ona toliko voli da priča.

Završila je nižu muzičku školu, a sada zajedno sa bratom svira sintisajzer. U disko ne voli da ide, a obožava da se mota po Knez Mihailovoj ulici. Skuplja konzerve, a sa svojim društvom ćaska skoro isključivo preko računara. U školi ima predmet informatiku, ali još ne zna kako s njom stoji jer ocena, za sad, nema.

Tamara je bratu predložila da krenu u biznis i otvore privatni BBS, ali Siniša nije siguran koliko bi taj posao bio unosan. Eto, to nam je ocutala Tamara, a ispričao njen brat. Kada sam stigla u redakciju, Tamara je već naveliko na „Sezamu" prepričavala utiske o ovom „razgovoru", a ja sam se gorko kajala što je nisam sačekala u zasedi na BBS-u. Sigurno bismo se site ispričale.

*Dada Vujasinović*

# ZVANIČNI INFORMACIONI SISTEM 29. ŠAHOVSKE OLIMPIJADE

## SEZAM COMMERCE

### NOVA ERA TRŽIŠNIH KOMUNIKACIJA

JEDINI MARKETINŠKI MEDIJ KOJI VAM PRUŽA

- **Dinamički marketing**
- **Interaktivnu komunikaciju**
- **Vlastitu reklamnu konferenciju**
- **Kontinuiranu promociju**
- **Ekspresno ispitivanje tržišta**
- **Stalni lični kontakt**

Sistem za modemske komunikacije – SEZAM – časopisa „Računari", jedinstven konferencijsko–informacioni sistem (danas radi na tri linije redno povezane na telefonski broj **011/647–955**, a do kraja godine broj linija će biti najmanje udvostručen), u svom marketinško–komercijalnom segmentu nudi vam mogucnosti kakve do sada niste imali

#### 1. ULAZNA ŠPICA

Reklamna poruka ne duža od deset reči pojavljuje se uz zaglavlje (špicu) Sezama pred svakim pretplatnikom svaki put kada uđe u sistem Klasični ekvivalent reklama u zaglavlju lista ili samoj špici TV dnevnika

#### 2. OSNOVNI MENI

Reklamna poruka, ne duza od tri reči koja se pojavljuje u osnovnom meniju To je centralna platforma sa koje se konsnik krece po čitavom sistemu Klasični ekvivalent sadrzaj i impresum u nekom časopisu

#### 3. OGLAS U KOMERCIJALNOM BLOKU

Reklamna poruka ne duža od 10 000 slovnih znakova (pet kucanih strana), emituje se u konferenciji OGLASI Oglašivač moze neograničeni broj puta da menja elemente poruke ili celu poruku Konterenciji OGLASI pretplatnici pnstupaju po sopstvenom izboru Klasični ekvivalent reklama u novinama ili između dve emisije na televiziji

#### 4 OGLAS U KONFERENCIJI

Reklamna poruka ne duža od 1 000 slovnih znakova (pola strane tekste) namenjena je konterenciji po izboru Oglašivač ne može da menja pojedine elemente poruke niti poruku kao celinu Radnim konterencijama pristupaju svi pretplatnici Klasični ekvivalent reklama u novinama u okviru teksta ili u prekidu emisije na televiziji

#### 5. ZAKUP KONFERENCIJE

Zakupom konferencije oglačivac moze da konsti odgovarajuci prostor na Sezamu prema svom nahođenju – za ostavljanje reklamnih poruka javnu komunikaciju sa pretplatnicima sistema ili što je posebno atraktivno, ze tehničku podršku svojih proizvoda pomoć i savete kupcima i povezivanje njihovih iskustava Klasični ekvivalent zakupljeni kanal na televiziji ili rediju sa non–stop emitovanjem i direktnim ukijučivanjem gledalaca odnosno slušalaca

#### 6. CIRKULARNA POŠTA

Reklamna poruke proizvoljne dužine automatski se odlaže u „poštansko sanduče" svakog pretplatnika ili određene grupe pretplatnika Poruka ima karakter lične pošte i potpuno je zaštićena Pretplatnik ako se to od njega očekuje i ako je ze to raspolozen moze da odgovori oglašivaču na reklamnu poruku, ovaj odgovor takođe ima karakter lične pošte i takođe je potpuno zašticen Klasični ekvivalent u drugim medijima ne postoji

#### 7 SPONZOR SISTEMA SEZAM

Sponzor sistema „Sezam" ima pravo, u trajanju od godinu dana na reklamnu poruku u špici sistema, na otvaranje sopstvene konterencije i neograničeni broj cirkularnih pisama – dakle na najkvalitetnije forme tržišnog komuniciranje ne Sezamu – kao i na jaku podršku u časopisu „Računari"

**SEZAM COMMERCE**

**ČETVRTI MEDIJ ZA ČETVRTU ERU TRŽIŠNIH KOMUNIKACIJA**

Tražite dopunske informacije i cenovnik!
Za sve informacije i dogovore – pozovite nas!
*Telefon* 011/653–748 • *Telefax* 011/648–140
*Telex* 11855 BIGZ YU • *BBS Sezam* 011/647–955

# MLAKAR & CO

**AUSTRIJA**

## Cenik programske opreme

Za savete kod izbora pozovite nas na telefon 9943/4227 2333. Naša trgovina je v Austriji u Podgori (Unterbergen) kraj glavnog puta prema Celovcu 60 km od Ljubljane i 12 km od Ljubelja. Trgovina je otvorena od 8 do 17 časova, a u soboto od 8 do 13 časova.

FAKS 9943/4227 2091 TELEKS 422749 MLCD A

| Borland | DM |
|---|---|
| Paradox 3 0 | 1 217 |
| Reflex 2 0 | 442 |
| SideKick Plus | 346 |
| Eureka | 285 |
| Turbo Assembler&Debugger | 259 |
| Turbo Basic | 181 |
| Turbo C Prof | 432 |
| Turbo C++ | 275 |
| Turbo Pascal | 275 |
| Turbo Pascal Prof | 432 |
| Turbo Prolog | 275 |
| Dualtro | 233 |
| Dualtro Pro | 890 |
| SuperKey | 181 |
| Lightning Word Wizard | 259 |
| Sprint Word Processor | 346 |
| Turbo Lightning | 181 |

| Microsoft | DM |
|---|---|
| MS Chart | 730 |
| MS Project | 888 |
| MS Works 2 0 | 259 |
| MS DOS v3 3 | 207 |
| MS DOS v4 1 | 207 |
| MS Windows/286 | 181 |
| MS Windows/386 | 338 |
| MS BASIC Compiler | 914 |
| MS C Compiler | 783 |
| MS COBOL Compiler v3 0 | 1 621 |
| MS FORTRAN Compiler | 909 |
| MS Macro Assembler | 275 |
| MS Pascal Compiler | 547 |
| MS Quick C Compiler | 181 |
| MS QuickBASIC | 181 |
| MS QuickC & Quick Assembler | 380 |
| MS QuickPascal | 233 |
| MS Excel | 353 |
| MS Multiplan | 800 |

| CAD | DM |
|---|---|
| Design CAD | 416 |
| AutoCAD V 10 | 6 545 |
| AutoSketch 2 0 | 249 |
| Drafix Architectural Symbols | 259 |
| Drafix Electric Symbols | 207 |
| Drafix Mechanical Symbols | 259 |
| Drafix Ultra | 878 |
| Auto Convert | 154 |
| Basic Home Design | 89 |
| Bathroom Design | 118 |
| CADD Level 3 W/Estimator | 730 |
| Commercial&Residential Design | 89 |
| Desk Convert | 89 |
| Electrical Engineering | 89 |
| Generic 3-D Drafting | 380 |
| Generic Cadd Level 3 | 442 |
| Generic Electronic Symbols I | 58 |
| Home Landscape | 58 |
| Steel Structures #1 | 110 |

| COMMUNICATION | DM |
|---|---|
| Crosstalk Xvi | 285 |
| Hot Wire | 198 |
| Procomm Plus | 154 |
| Smartcomm III | 390 |
| Lotus Express | 285 |
| Carbon Copy Plus | 301 |
| SmarTerm 400 | 275 |
| LapLink III | 249 |
| Viewlink | 233 |

| DATABASE | DM |
|---|---|
| RapidFile | 573 |
| dBase IV v 1 1 | 1 280 |
| dBase IV Developers Kit | 2 301 |
| Clarion Personal Developer | 327 |
| DataFlex | 861 |
| DataEase | 1 306 |
| Foxbase Plus | 511 |
| Foxpro | 1 228 |
| R BASE for DOS | 1 280 |
| Clipper 5 0 | 1 165 |
| Clipper Summer 87 | 1 045 |
| Btrieve | 416 |
| ORACLE Professional | 2 356 |
| SCO FoxBASE+ 286 | 1 725 |
| Data Perfect | 809 |

| DESKTOP PUBLISHING | DM |
|---|---|
| Page Maker | 1 280 |
| Byline | 511 |
| Publisher Powerpack (FP) | 128 |
| Books & Manual Collections | 416 |
| Newsletter Collections | 416 |
| Omnipage | 3 793 |
| Corel Draw | 988 |
| Gem/3 Desktop Publisher | 458 |
| PFS First Publisher | 223 |
| Publish It! | 312 |
| Ventura Publisher 3 0 Golden | 1 411 |

| GRAPHICS | DM |
|---|---|
| Pizazz Plus | 196 |
| Applause II | 170 |
| ChartMaster | 573 |
| DiagramMaster | 573 |
| MapMaster | 878 |
| Master Graphics Present | 809 |
| Autodesk Animator | 878 |
| Arts & Letters Composer | 704 |
| Gem/3 Draw Plus | 469 |
| Gem/3 Graph | 469 |
| Story Board Plus | 600 |
| Inset + Hijaak | 364 |
| Lotus Freelance Plus | 888 |
| Lotus Graphwriter II | 888 |
| Graph Plus | 835 |
| Db Graphics | 583 |
| Harvard Graphics | 881 |
| Pixie | 312 |

| INFORMATION PROJECT | DM |
|---|---|
| Project Workbench Adv System | 2 144 |
| Super Project Expert | 1 175 |
| Gem/3 Desktop | 82 |
| Lotus Agenda | 730 |
| Harward Project Manager | 1 229 |
| Plan Perfect | 678 |
| Wordperfect Library | 181 |
| Wordperfect Office | 730 |

| INTEGRATED | DM |
|---|---|
| Framework III | 1 217 |
| Lotus Symphony 2 2 | 1 359 |

| MATHEMATICAL | DM |
|---|---|
| Math Cad 2 5 | 783 |
| Statgraphics | 1.542 |
| Systat | 1 139 |
| Mathematica for DOS 386 | 2.088 |

| NETWORKING | DM |
|---|---|
| Adv Netware 286 V2 15 | 4 631 |
| ELS NetWare Level I | 1 073 |
| ELS NetWare Level II | 2 589 |
| Netware 386 | 12 799 |
| SFT V2 15 | 7 563 |
| XDL | 1 333 |

| OPERATING SYSTEMS | DM |
|---|---|
| DESQview | 207 |
| DESQview | 353 |
| SCO XENIX 286 Complete System | 2 484 |
| SCO XENIX 286 Develop System | 1 202 |
| SCO XENIX 286 Operating System | 1 296 |
| SCO XENIX 386 Complete System | 2 746 |
| SCO XENIX 386 Develop System | 1 437 |
| SCO XENIX 386 Operating System | 1 228 |

| PROGRAMMING | DM |
|---|---|
| C TOOLS PLUS | 327 |
| POWER TOOLS PLUS | 285 |
| Turbo C TOOLS Plus | 285 |
| Smalltalk/V | 207 |
| Lattice C Compiler | 442 |
| Pro-C | 704 |
| Zortech C++ | 406 |

| SPREADSHEETS | DM |
|---|---|
| SuperCalc V | 835 |
| Lucid 3D | 182 |
| Allways | 233 |
| Sideways | 110 |
| Worksheet Utilities | 162 |
| Lotus 1 2-3 V 3 0 | 1 045 |
| Lotus HAL | 285 |
| Vp Planner 3D | 408 |
| PFS Professional Plan | 182 |
| Note-It Plus | 154 |
| SQZ! Plus | 207 |

| UTILITIES | DM |
|---|---|
| Above Disc | 181 |
| Personal Music Composer | 390 |
| Popdrop | 92 |
| Disk Vision | 128 |
| Copy II PC | 65 |
| PC Tools Deluxe V 6 0 | 233 |
| CoreFast | 198 |
| Laser Control | 259 |
| Lotus Magellan | 312 |
| Lotus Metro | 170 |
| Auto Menu | 110 |
| Norton Advanced Utilities | 233 |
| Norton Back Up | 233 |
| Norton Commander | 199 |
| Norton Editor | 128 |
| Disk Explorer | 170 |
| 386 Max | 144 |
| Manifest | 102 |
| QEMM 386 | 162 |
| QRAM | 144 |
| Disk Optimizer | 118 |
| Liberty | 469 |
| xTree Pro | 207 |

| WORD PROCESSING | DM |
|---|---|
| Multimate | 772 |
| Notabene | 704 |
| Display Write IV 2 0 | 704 |
| Lotus Manuscript | 903 |
| Wordstar 2000+ | 730 |
| Wordstar 6 0 | 547 |
| Grammatic IV | 144 |
| Steadmans Medical Dictionary | 154 |
| PFS Professional Write | 390 |
| WordPerfect 5 1 | 878 |

**Za programsku opremu koja u cenovniku nije navedena, budite ljubazni pa nas pozovite.**

# MLAKAR & CO

AUSTRIJA

| KUĆIŠTA SA UREĐAJEM ZA NAPAJANJE | DEM | DIN |
|---|---|---|
| AT BABY | 128 | 1 254 |
| MINI TOWER | 251 | 2 464 |
| TOWER | 311 | 3 046 |

| OSNOVNE PLOČE | | |
|---|---|---|
| XT 4.77 10 MHz | 115 | 1 129 |
| AT 286 12MHz | 180 | 1 764 |
| NEAT 286 16MHz | 280 | 2 744 |
| 386 SX 16 | 665 | 6 517 |
| 386 25MHz | 1 120 | 10 976 |
| 386 25MHz 64 K CACHE | 1 730 | 16 954 |
| 386 33MHz 64 K CACHE | 1 990 | 19 502 |
| 486 25 MHz | 3 900 | 38 220 |

| DISPLAY KARTICE | | |
|---|---|---|
| Printer/Hercules | 38 | 372 |
| VGA 800×600/8 bit | 129 | 1 264 |
| Super VGA 1024×768 | 210 | 2 058 |

| KONTROLERI | | |
|---|---|---|
| HDD XT MFM | 96 | 941 |
| FDD HDD AT MFM 1:1 | 130 | 1 274 |
| DTC 7280 AT MFM 1:1 | 210 | 2 058 |
| DTC 7287 AT RLL 1:1 | 270 | 2 646 |
| AT(IDE) BUS FDD/HDD | 44 | 431 |
| SCSI FDD/HDD | 96 | 940 |
| ESDI FDD HDD | 280 | 2 744 |

| DODATNE KARTICE | | |
|---|---|---|
| MULTI IO XT | 69 | 672 |
| I O AT (SER PORT) | 28 | 273 |
| I O AT (PAR SER PORT) | 34 | 333 |
| RAM CARD EMS 2MB AT | 170 | 1 667 |

| LAN | | |
|---|---|---|
| Ethernet compat (NE1000) 8 8bit | 235 | 2 303 |
| Ethernet compat (NE2000) B 16bit | 280 | 2 744 |
| Ethernet boot rom for NE1000 | 10 | 100 |
| Ethernet boot rom for NE2000 | 10 | 100 |
| Ethernet IEEE802 3 transceiver piercing | 314 | 3 080 |
| Ethernet IEEE802 3 transceiver n type | 226 | 2 219 |
| Ethernet IEEE802 3 transceiver BNC | 212 | 2 086 |
| BNC 50 ohm terminator | 6 | 56 |
| BNC 93 ohm terminator | 6 | 56 |
| N series 50 ohm female terminator | 9 | 84 |
| Cable RG 58 (1M) | 3 | 29 |
| Cable connector | 6 | 56 |
| Ethernet IEEE802 3 repeater | 1 207 | 11 830 |
| Arcnet coax star LAN card | 125 | 1 232 |
| Arcnet coax bus LAN card | 138 | 1 358 |
| Arcnet twisted pair star LAN card | 138 | 1 358 |
| 4 port coax at active hub card | 314 | 3 080 |
| 4 port twisted pair hub card | 378 | 3 710 |
| Remote boot rom for arcnet card | 10 | 100 |
| Cable RG 62 (1M) | 3 | 29 |

| TASTATURE | | |
|---|---|---|
| 102 tipk | 58 | 568 |
| 102 tipke click Chicony | 78 | 764 |
| 101 tipka click Chicony YU | 95 | 931 |
| 101 tipka z miško Chicony | 167 | 1 640 |
| 101 tipka Cherry | 138 | 1 352 |

| GIBKI DISKOVI | | |
|---|---|---|
| 5.25" 360 Kb | 121 | 1 186 |
| 5.25" 1.2 Mb | 125 | 1 225 |
| 3.5" 1.44 Mb | 125 | 1 225 |

| TVRDI DISKOVI | | |
|---|---|---|
| Seagate 20 Mb 65 ms | 380 | 3 724 |
| Seagate 45 Mb/28 ms AT BUS | 500 | 4 900 |
| Seagate 85 Mb 28 ms SCSI | 690 | 6 762 |
| NEC 135 Mb 23 ms ESDI | 1 764 | 17 290 |
| NEC 179 Mb/18 ms ESDI | 1 950 | 19 110 |

Računare prodajemo u KIT verziji (u delovima). Za sve uređaje nudimo garanciju, montažu, servis u Jugoslaviji. Za savete kod izbora pozovite nas na telefon 9943 4227 2333. Naša trgovina je v Austriji u Podgori (Unterbergen), kraj glavnog puta prema Celovcu, 60 km od Ljubljane i 12 km od Ljubelja. Trgovina je otvorena od 8 do 17 časova, a u soboti od 8 do 13 časova.

FAKS 9943 4227 2081 TELEKS 422749 MLCO A



## IBM KOMPATIBILNI RAČUNARI

| | DEM |
|---|---|
| **XT 10-21** | **999** |
| XT 4.77/12 MHz 640 K RAM 20 Mb trdi disk 14" monokr. monitor | |
| **AT 286-12-45** | **1.392** |
| AT 286-12 MHz 1MB RAM 45 Mb trdi disk 14" monokr. monitor | |
| **AT 286-NEAT-16-45** | **1.510** |
| NEAT 286 16 MHz 1Mb RAM 45 Mb trdi disk 14" monokr. monitor | |
| **AT 386 SX-45** | **1.895** |
| AT 386 SX 1 Mb RAM 45 Mb trdi disk 14" monokr. monitor | |
| **AT 386-25-45** | **2.473** |
| AT 386 25 MHz 1 Mb RAM 45 Mb trdi disk 14" monokr. monitor | |
| **AT 486-25-45** | **5.410** |
| AT 486 25 MHz 1 Mb RAM 45 Mb trdi disk 14" monokr. monitor | |

**Garancija 24 meseci**

Mlacom

MLACOM d.o.o.
Celovška 185
61000 Ljubljana 1
Tel. 061/556-484
Fax 061/556-485

2270-395

| MONITORI | | |
|---|---|---|
| 14" monokromatski | 189 | 1 852 |
| VGA Color 1024 × 768 | 710 | 6 958 |
| 15" A4 full size VGA | 1 540 | 15.092 |
| NEC Multisync 2A | 1 050 | 10 290 |
| NEC Multisync 3D | 1 350 | 13 230 |
| NEC Multisync 5D | 4 980 | 48.804 |

| ŠTAMPAČI | DEM | DIN |
|---|---|---|
| CITIZEN 180D A4 180 zn/sek | 350 | 3 430 |
| C T I 9 Pin A3 | 684 | 6 703 |
| Star LC 15 | 745 | 7 301 |
| Star LC 24-10 | 645 | 6 321 |
| Star LC 24-15 | 1 010 | 9 898 |
| Laser HP JET II P | 2 300 | 22 540 |
| Laser HP JET III | 3 890 | 38 122 |

| CRTAČI | | |
|---|---|---|
| ROLAND DXY 1100 A3 | 1 690 | 16 562 |
| ROLAND DXY 1200 A3 | 2 421 | 23 730 |
| ROLAND DXY 2200 A2 | 6 605 | 64 736 |

| MODEMI | | |
|---|---|---|
| 2400 int | 233 | 2 285 |
| 2400 ext | 274 | 2 688 |

| UPS – NEPREKIDNO NAPAJANJE | | |
|---|---|---|
| UPS 300 VA | 480 | 4 704 |
| UPS 500 VA | 570 | 5 586 |
| UPS 1000 VA | 1 097 | 10 752 |

| RAM | | |
|---|---|---|
| 4164-10 | 3 | 29 |
| 41256-10 | 4 | 39 |
| 41256-08 | 4.50 | 44 |
| 44256-08 | 16 | 157 |
| 411000-08 | 16 | 157 |

| COPROCESSOR | | |
|---|---|---|
| 8087-1 (10MHZ) | 410 | 4 018 |
| 80287-10MHZ | 455 | 4 459 |
| 80387SX 16MHZ | 690 | 6 762 |
| 80387-20MHZ | 940 | 9 212 |
| 80387-25MHZ | 1 030 | 10 094 |

| STREAMER | | |
|---|---|---|
| COLORADO 40/60/120 Mb int | 795 | 7 791 |
| COLORADO 40/60/120 Mb ext | 1 150 | 11 270 |

| RAZNO | | |
|---|---|---|
| Čitalnik črtne kode | 466 | 4 569 |
| Prenosni čitalnik črtne kode | 1 042 | 10 214 |
| Miška Genius 6-Plus | 68 | 666 |
| Miška Genius GM F-302 | 89 | 870 |
| Tablet Genius GT 1212B 12 X 12 | 532 | 5 214 |
| Scanner Handy Geniscan GS 4500 | 296 | 2 901 |
| Scanner A4 Handy w/pap feeder | 1 600 | 15 680 |
| Eprom UV Eraser | 260 | 2 548 |
| Eprom Writer Card 4x | 392 | 3 850 |
| Disk Box 5 × 5.25 | 2 | 21 |
| Disk Box 10 × 5.25 | 4 | 35 |
| Disk Box 50 × 5.25 | 12 | 117 |
| Disk Box 5 × 3.5 | 3 | 29 |
| Disk Box 10 × 3.5 | 3 | 33 |
| Copy Holder | 14 | 138 |

DIN su cene bez poreza na promet kod Mlacom Ljubljana

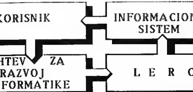
DEM su cene bez poreza na promet kod Mlakar & CO Austrija

**Na zalihi i druga oprema**

**LERC** LESKOVAC

16000 LESKOVAC
AVNOJA 95
TELEFON 016/ 46-512
46-317
TELEFAX 016/ 48-117

## INTEGRISANI KONZISTENTNI VISEKORISNICKI INFORMACIONI SISTEMI




- AUTOMATIZOVAN NORMALIZOVAN POSLOVNI INFORMACIONI SISTEM
  - Finasijsko knjigovodstvo, Robno-Materijalno knjigovodstvo,
  - Licni dohodak, Osnovna sredstva, Potrosacki krediti
  - Analiticka i sinteticka resenja po zahtevu korisnika
- POSLOVNI SISTEMI ZDRAVSTVA
  - Obrada recepata
  - Pracenje eksterne potrosnje
  - Obrada bolovanja
  - Jedinstvena evidencija osiguranika
  - Bolnicko i Apotekarsko poslovanje
- POSLOVNI SISTEM BANKARSKOG POSLOVANJA

**IZRADA EKSPERTNIH SISTEMA PREMA ZAHTEVIMA KORISNIKA NA NIVOU OPERATIVNOG SISTEMA I SOFTVERSKOG ALATA PO IZBORU KORISNIKA**

- EKSPERTNI SISTEM ZA MEDICINSKU DIJAGNOSTIKU
  - Nacin i podrucje primene po izboru Lekara
- ESPERTNI SISTEMI I PROGRAMSKI PAKETI ZA SIMULACIJE MEDICINSKIH PROCESA
  - Program za simulaciju rada SRCA sa mogucnoscu merenja efekata vremenskih intervala fibrilacionih impulsa
- PRIMENA RUCNIH-DZEPNIM RACUNARA U ZDRAVSTVU

**OSNOVNI ALATI : O.S. UNIX , XENIX , QNX , NOVELL , PS-MOS , MS DOS**
**RELACIONA BAZA , ALAT IV-GENERACIJE : ZIM**

MJEŠOVITO PREDUZEĆE **GENELEC**

ELEKTRONIKA & INFORMATIKA, proizvodnja — servis — promet

75000 TUZLA
Zlokovac 3B
tel. 075/231-557
fax. 075/237-778

# NE, VI NE SANJATE!
# OVO JE SAN KOJI VAM SE UPRAVO DOGAĐA

## PONUDA RAČUNARA

| za MS DOS | | za SCO UNIX/XENIX | |
|---|---|---|---|
| **XT 12 MHz/32 MB**<br>baby XT 4 77/12 MHz 640 KB<br>32 MB HDD, ser + paralel,<br>c/b monitor, YU znaci | **18.737 DIN** | **386 XENIX — 4 USER**<br>Mini Tower 386/25 MHz, 2 MB RAM WD MFM FDD/HDD Contr 1 1,<br>50 MB HDD 40 MB Streamer Flynix 4 serial port card<br>ser + paralel port c/b monitor YU znaci SCO XENIX 386 | **64 394 DIN** |
| **AT 12 MHz/45 MB**<br>baby 286/12 MHz, 1 MB 45 MB<br>HDD (1 1) ser + paralel<br>c/b monitor, YU znaci | **17.987 DIN** | **386 XENIX — 8 USER**<br>Tower 386/25 MHz 4 MB RAM, WD RLL FDD/HDD Contr 1 1<br>120 MB HDD 60 MB Streamer Flynix 8 serial port card<br>ser + paralel port c/b monitor YU znaci SCO XENIX 386 | **77.079 DIN** |
| **NEAT 16 MHz/45 MB**<br>baby, NEAT 286/16 MHz 1 MB,<br>45 MB HDD (1 1) ser + paralel,<br>c/b monitor YU znaci | **20 222 DIN** | **386 XENIX — 16 USER**<br>Tower 386/25 Cache 64 KB 4 MB RAM WD ESDI FDD/HDD Contr 1 1,<br>Maxtor 150 MB HDD 60 MB Streamer Specialix 16 ser port card<br>ser + paralel port c/b monitor, YU znaci SCO XENIX 386 | **122.993 DIN** |
| **386 SX 16 MHz/45 MB**<br>baby 386 SX/16 MHz, 1 MB<br>45 MB HDD (1 1) ser + paralel<br>c/b monitor, YU znaci | **22.868 DIN** | **386 XENIX — 32 USER**<br>Tower 386/33 Cache 64 KB 8 MB RAM WD ESDI FDD/HDD Contr 1 1,<br>Maxtor 338 MB HDD 150 MB Streamer Specialix 32 ser port card<br>ser + paralel port c/b monitor, YU znaci SCO XENIX 386 | **170.524 DIN** |
| **386 20 MHz/85 MB**<br>Mini Tower, 386/20 MHz, 2 MB<br>85 MB HDD (SCSI) ser + paralel<br>c/b monitor, YU znaci | **31 351 DIN** | **386 XENIX — 64 USER**<br>486 — Tower 486/25 16 MB RAM, WD ESDI FDD/HDD Contr 1 1,<br>Maxtor 676 MB HDD 150 MB Str 2 X Specialix 32 ser port card<br>ser + paralel port c/b monitor, YU znaci SCO XENIX 386 | **288.205 DIN** |

## PONUDA DIJELOVA

| KUĆIŠTA | DIN |
|---|---|
| Slim Line 8 PS 150 W | 2.667 |
| Baby 8 PS 200 W | 1 906 |
| Mini Tower 8 PS 200 W | 2 396 |
| Tower 8 PS 230 W | 4 184 |
| 486 Tower 8 PS 250 W (sa 2 ventilatora) | 8 102 |

| MATIČNE PLOČE | |
|---|---|
| XT 4 77/12 MHz | 1 205 |
| AT-286/12 MHz | 2 343 |
| AT 286/16 MHz NEAT | 4.283 |
| AT 286/20 MHz NEAT | 5 354 |
| 386 SX/16 MHz | 6 960 |
| 386 SX/20 MHz | 8 992 |
| 386/20 MHz | 12 716 |
| 386/25 MHz | 14 992 |
| 386/25 MHz Cache 64 KB | 18 205 |
| 386/33 MHz Cache 64 KB | 22 758 |
| 486/25 MHz | 53 543 |

| I/O KARTICE | |
|---|---|
| XT Multi I/O | 723 |
| AT I/O (ser + paralel) | 295 |
| Paralel port card | 218 |
| 2 X RS 232C card | 335 |

| KONTROLERI (sa kablovima) | DIN |
|---|---|
| XT MFM WD GEN | 1 178 |
| XT RLL WD 1004 A-27X | 1 245 |
| AT MFM WD 1003V MM2 (2 1) | 2 209 |
| AT RLL WD 1003V-SR2 (2 1) | 2 409 |
| AT MFM WD 1006V MM2 (1 1) | 2 342 |
| AT MFM WD 1006V MM2 (1 1) | 2 610 |
| AT BUS (IDE) Adapter 8 FDC | 469 |
| ESDI WD 1007V-SE2 (1 1) | 4 083 |
| SCSI WD 7000 FASST | 8 366 |
| Seagate ST-02 SCSI | 998 |

| DISKETNI POGONI | |
|---|---|
| 5 25" 360 KB | 1 178 |
| 5 25" 1 2 MB | 1.339 |
| 3 5" 1 44 MB bez rama 5 25" | 1 178 |
| 3 5" 1 44 MB sa ramom 5 25" | 1 312 |

| TVRDI DISKOVI SEAGATE | | |
|---|---|---|
| Seagate ST 225 | 20 MB/65 ms | 3 989 |
| Seagate ST 238R | 32 MB/65 ms | 4 390 |
| Seagate ST 251-1 | 40 MB/28 ms | 5.354 |
| Seagate ST 157A-1 | 45 MB/28 ms | 5 555 |
| Seagate ST 277R 1 | 66 MB/28 ms | 5 689 |
| Seagate ST 4096 | 80 MB/28 ms | 12 315 |
| Seagate SCSI | 85 MB/28 ms | 7 228 |

| OSTALI TVRDI DISKOVI | DIN |
|---|---|
| Maxtor 4170 ESDI/SCSI 150 MB/16 ms | 25 386 |
| Maxtor 4380 ESDI/SCSI 338 MB/18 ms | 37 346 |
| Maxtor 8380 ESDI/SCSI 338 MB/16 ms | 38.818 |
| Maxtor 8760 ESDI/SCSI 676 MB/18 ms | 55 551 |
| Micropolis 1654 7 ESDI 161 MB/18 ms | 22 622 |
| Micropolis 1664 7 ESDI 340 MB/14 ms | 32 795 |
| Micropolis 1674-7 SCSI 161 MB/18 ms | 24 084 |
| Micropolis 1684-7 SCSI 340 MB/14 ms | 38 150 |

| STRIMERI WANGTEK | |
|---|---|
| 3040 FK, 40 MB int 5 25/3 5" | 6 680 |
| 5099 EK 60 MB int 5 25 | 12 650 |
| 5150 PK 150 MB, int. 5 25 | 16.685 |

| MEMORIJE | |
|---|---|
| DRAM 4164-10 | 39 |
| DRAM 41464 10 | 49 |
| DRAM 41256-10 | 32 |
| DRAM 44256-10 | 118 |
| DRAM 411000-07 | 117 |
| SIMM/SIP MODUL 256*9 10 | 381 |
| SIMM/SIP MODUL 1 M*9-07 | 1 165 |
| EPROM 2764/200 | 71 |
| EPROM 27512-150 | 110 |
| EPROM 271000-150 | 306 |

| KOPROCESORI | DIN |
|---|---|
| 8087 1 | 3 350 |
| 80287 10 | 4 752 |
| 80387SX 16 | 6 559 |
| 80387 20 | 8 567 |
| 80387 25 | 10 307 |
| 80387-33 | 12 582 |

| GRAFIČKE KARTICE | |
|---|---|
| Hercules komp | 398 |
| VGA/16 800×600 | 2 249 |
| SuperVGA 1024×768 | 2 998 |

| MONITORI | |
|---|---|
| 14" VGA Mono 640×480 | 3 197 |
| 14" VGA Mono 1024×768 | 5 676 |
| 14" Visa MC 6720 | 11 110 |
| 14" NEC 2A | 12 718 |
| 14" NEC 3D | 18 732 |

| OSTALO | |
|---|---|
| Miš Quick DM E1 | 708 |
| Tablet Genius GT-1212B | 6 680 |
| Modem Discovery 1200 | 1 740 |
| Modem Discovery 2400 | 2 811 |

A NA KRAJU POSLASTICA matrični laserski i linijski štampač DATAPRODUCTS vrhunskih performansi i kvaliteta! POZOVITE NAS RADI CIJENA

## Microline AT 16/40

Takt 16MHz RAM 1 MB hard disk 40 MB 25 ms 3 5' floppy 1 2 ili 1 44 MB Hercules kartica i monitor

**17 600,-**

## Microline 386SX 16/60

Takt 16 MHz, RAM 1 MB, hard disk 64 MB 28 ms, floppy 1 2 ili 1 44 MB Hercules kartica i monitor

**23 900,-**

## Microline 386 25/100

Takt 25 MHz RAM 4 MB, hard disk 100 MB 25 ms 3 5', floppy 1 2 ili 1 44 MB Hercules kartica i monitor, mini tower kućište

Best buy

**41 000,-**

## Microline 386 33/100

Takt 33 MHz 64 KB cache RAM 4 MB hard disk 100 MB 25 ms 3 5', floppy 1 2 MB, Hercules kartica i monitor, tower kućište

**49 000,-**

## Microline 486 25/200

Takt 25 MHz, 128 KB cache, RAM 4 MB hard disk 200 MB 16 ms 3 5', floppy 1 2 MB, Hercules kartica i monitor, tower kućište

**98 000,-**

## Microline 486 Server

Takt 25 MHz, 128 KB cache, RAM 16 MB, hard disk 330 MB 11 ms, floppy 1 2 MB, Hercules kartica i monitor, tower kućište

**135 000,-**

U računala ugrađujemo hard diskove Conner CDC i Miniscribe floppy diskove TEAC i Y-E Data monitore u boji NEC i EIZO te kontrolere Western Digital i OMTI Svako računalo ima serijski, paralelni i game port te klik tastaturu i YU set Prodajemo i FISA sisteme, cijene na upit

## Printeri

| | |
|---|---|
| EPSON LX400 | 5 180 |
| EPSON LX850 | 6 400,- |
| EPSON FX850 | 14 200,- |
| EPSON FX1050 | 14 150 |
| EPSON LQ550 | 11 390, |
| EPSON LQ850 | 16 800,- |
| EPSON LQ860 | 25 400,- |
| EPSON LQ1010 | 17 000,- |
| EPSON LQ1050 | 22 300,- |
| EPSON LQ1060 | 31 300 - |
| EPSON LQ2550 | 40 400,- |
| EPSON DFX5000 | 48 800,- |
| EPSON DFX8000 | 73 000,- |
| HP IIP, laser | 28 000,- |
| HP III, laser | 44 800,- |
| HP IID, laser | 65 600 - |
| RAM 1 MB za HP | 4 000 - |
| RAM 2 MB za HP | 5 500,- |
| RAM 4 MB za HP | 8 500 - |
| Kabel za printer | 200 - |

Best buy

# Microline

Sjedište Zagreb, Jordanovec 119

Prodaja i servis Štoosova 25

Tel 041/217-915, fax 041/216-711

## Mreža

Želite li poslovanje Vašeg poduzeća osuvremeniti? Smatrate li da je ažurna informacija važna svakoj odgovornoj osobi u cijelom poduzeću? Više ne morate koristiti skupe i spore usluge velikih ERC-ova ni stalno gubiti vrijeme nego instalirajte suvremene Novell mreže. Vrijeme odziva se više neće mjeriti satima ni danima nego sekundama uz manju cijenu i veću sigurnost.




| | |
|---|---|
| RPTI 1 MB/s | 5 000 /čvor |
| RPTI, 10 MB/s | 7 000 /čvor |
| Novell ELS II | 25 000,- |
| Novell 286 2 15 | 40 000 - |
| Novell 286 2 15 SFT | 64 000 - |
| Novell 386 | 105 000 - |
| Novell hardware | 4000 /čvor |

## Matične ploče

| | |
|---|---|
| AT 16 MHz | 2 670 |
| 386SX 16 MHz | 6.490, |
| 386 25 MHz | 14 360 - |
| 386-33, 64K cache | 21 400 - |
| 486-25, 128K cache | 61 800,- |

Best buy

## RAM čipovi

| | |
|---|---|
| 41256-100 | 25,- |
| 44256-80 | 99, |
| 411000-80 | 99,- |
| 4464-100 | 40, |
| SIP & SIMM 9*M 80 | 1 100 - |

Best buy

## Floppy diskovi

| | |
|---|---|
| TEAC, YE 1 2 MB | 1 360 |
| TEAC 1 44 MB | 1 280 - |
| TEAC 360 KB | 990, |
| Rama za 3 5' floppy | 190, |

Best buy

## Hard diskovi

| | |
|---|---|
| Mitsubishi 64M 28ms | 7 000, |
| Conner 200M 16ms | 21 300, |
| Conner 100M 25ms | 13 500,- |
| Conner 40M 25ms | 7 500, |
| Miniscribe 40M 28ms | 6 300, |

## Hard disk kontroleri

| | |
|---|---|
| MFM OMTI | 1 240 - |
| RLL OMTI | 1 530, |
| RLL 256K cache, OMTI | 1 870 - |
| SCSI , OMTI | 2 070 - |
| ESDI OMTI | 3 970 - |
| AT bus, OMTI | 500 - |
| AT bus + multi I/O | 600 |

Best buy

Svi kontroleri su AT kombi i is ... U cijenu su uračunati kablovi Uz OMTI kontrolere prodajemo DOS utilities za streamer WORM i R/W diskove, Novell, UNIX i OS/2 drivere i utilities.

## Grafičke kartice

| | |
|---|---|
| Hercules YU+sklopka | 405,- |
| VGA 512 KB TSENG | 2 700,- |
| VGA 1 MB Trident | 2 900,- |

## Miševi

| | |
|---|---|
| GM F302 | 970,- |

## Multi I/O

| | |
|---|---|
| AT I/O 2S+P+G | 340,- |
| XT multi I/O | 400,- |
| IEEE 488 | 2 850,- |

## Modemi

| | |
|---|---|
| Modem 2400 bauda | 1 800,- |

## Ethernet

| | |
|---|---|
| Ethernet kartice, 8 bitna | 2 150,- |
| Ethernet kartice, 16 bitna | 2 600, |

## Kućišta s napajanjem

| | |
|---|---|
| AT flip top | 1 400,- |
| AT slim line | 1 800, |
| Mini tower | 2 450,- |
| Tower | 3 500 - |

## Testature i monitori

| | |
|---|---|
| Tastatura | 1 130,- |
| Hercules monitor | 2 290,- |
| VGA 1024*768 (CTX) | 9 270,- |
| NEC 2A | 14 210,- |
| NEC 3D | 17 780,- |
| EIZO 8070S | 26 200,- |

## Diskete 3M

| | |
|---|---|
| 5 25' DS/DD 10 kom | 207 - |
| 3 5' DS/DD, 10 kom | 350,- |

## Streameri

| | |
|---|---|
| 60 MB, interni | 7 650, |
| 60 MB int + kontroler | 8 960 - |
| 150 MB interni | 16 930 - |

## Koprocesori

| | |
|---|---|
| Intel 80287-8 | 4 360,- |
| Intel 80287-10 | 4 760 - |
| Intel 80287XL 12 | 4 400, |
| Intel 80387SX 16 | 6 830, |
| Intel 80387SX-20 | 7 720, |
| Intel 80387-16 | 7 550,- |
| Intel 80387 20 | 6 680 - |
| Intel 80387-25 | 10 970,- |
| Intel 80387-33 | 13 410,- |
| IIT 2C87-8 | 3 940 - |
| IIT 2C87-10 | 4 190 - |
| IIT 2C87-12 | 4 190, |
| IIT 2C87 20 | 5 270,- |
| IIT 3C87SX 16 | 6 330 - |
| IIT 3C87-16 | 6 610,- |
| IIT 3C87 20 | 6 020,- |
| IIT 3C87-25 | 10 020,- |
| IIT 3C87-33 | 12 250 |
| Cyrix 83D87SX 16 | 6 550,- |
| Cyrix 83D87 16 | 6 300 - |
| Cyrix 83D87 20 | 9 390, |
| Cyrix 83D87-25 | 11 770,- |
| Cyrix 83D87 33 | 12 450 - |

Best buy

Cyrix koprocesori su otprilike 3 puta brži od Intel-a

| | |
|---|---|
| Weitek 3167-25 | 17 780,- |
| Weitek 3167-33 | 23 170, |
| Weitek 4167 25 | 24 960, |
| Weitek 4167 33 | 32 110,- |

Novo

Na skladištu se ne nalaze sve vrste koprocesora Ukoliko traženog modela nema na skladištu rok isporuke je oko 15 dana

## EPROMi

| | |
|---|---|
| 2764 25 | 54 |

## MS DOS

| | |
|---|---|
| DOS4 01 + GW Basic | 1 100,- |

Garancija 12 mjeseci Cijene važe za poduzeće i u njih nije uračunat porez na promet. Sve cijene su FCO Zagreb, Štoosova 25, ali je dostava računala u Zagrebu besplatna Minimalna vrijednost za slanje robe iznosi 7 000 din Rok za isporuku 0-30 dana


Beohard®
Beohard u saradnji sa "MULTI COMERC" d.o.o.
Vam nudi po izuzetno povoljnim cenama :
Commodore 64/II
"Najbolji odnos perfomansa-cena" časopis Računari
samo
3399,-
commodore kasetofon
palica za igru-quickshoot II
ispravljač
svi neophodni kablovi
kaseta sa 10 super igara
prevod uputstva
garancija 12 meseci
obezbeđen softver
Amiga 500
Proglašen za "računar godine" časopisi Byte, Svet kompjutera ..
samo
9100,-
RF modulator za TV
miš (mouse)
ispravljač
svi neophodni kablovi
2 diskete (Work Bench & Amiga Basic)
1 disketa sa igrama
prevod uputstva
garancija 12 meseci
obezbeđen softver
Garancija za ava naše usiuge je 12 maseci
Narudžbina primamo taiafonom ili putam pisma
Plaćanje je pouzećem - poštaru kada Vam donese računar
PTT troškova i troškove pakovanja snosi kupac
Računare po dogovoru donosimo kući
Prako 5 kupljenih računara odobravamo popust od 8%
Narućite odmah
Beohard, Post.fah 78, 11050 Beograd 22 ☎ 011/420-661
vreme od 8h do 21h svakim danom i subotom i nedeljom

# MOĆ UNIX-A

# SNAGA i486

ALTOS nudi tri osnovne familije sistema.

| 386/25 System 600 | 386/33 System 1000 | 486/25 System 1000 |
|---|---|---|
| Intel 80386, 25 MHz | Intel 80386, 33MHz | Intel 80486, 25Mhz |
| 32 kB cache memorija | | 8 kB cache na čipu |
| 4—8 MB memorije | 4—24 MB memorije | 4—28 MB memorije |
| 90—380 MB disk | 190—700 MB disk (do 2,5 GB sa SEU) | 170—630 MB disk (do 4,9 GB sa SEU) |
| 125/150 MB streamer traka | | |
| 1,2 MB 5,25" floppy | | |
| 8 RS 232 linija | 8—24 RS 232 linija (do 72 linije sa multidropom) | (do 128 linija sa multidropom) |
| do 8 korisnika | do 32 korisnika | do 64 korisnika |

Standardizovani hardver omogućava prelaz između familija jednostavnom zamenom



**kma** D.O.O.

Milentija Popovića 9
BEOGRAD (Sava centar)
Tel. 011/222-43-22, lok. 448
Fax 011/455-785

**REŠENJE TESTA SA STR 60**

## ČESTITAMO, VI STE GENIJE

| | |
|---|---|
| **41+:** | Vi ste savršenstvo bez mane. Ipak, možda ne bi bilo loše da upišete kurs elementarne aritmetike |
| **38-40:** | Fantastičan rezultat. Ipak, da li ste sigurni da niste negde odstupili od pravila testa? |
| **35-38:** | Sjajan rezultat. Vrata bilo koje računarske institucije su vam širom otvorene |
| **31-35:** | Odličan rezultat. Veliki potencijal. Ipak, tu i tamo, pokušajte da još ponešto naučite |
| **24-30:** | Solidno znanje. Mesta za dalji napredak imate, ali u principu možete biti zadovoljni sobom |
| **16-23:** | Možda talenta imate, ali sigurno da ne bi bilo loše da pogledate još po koju knjigu |
| **9-15:** | OK, zaboravite na računarstvo, upišite pravo ili političke nauke |
| **1-8:** | Pretpostavljam da ste ili izuzetno atraktivna gospođica ili da ste nasledili izuzetno atraktivne pare. Ako je tako, niste ni trebali gubiti vreme sa testom. Ali, ako nije, šta onda, pitanje je sad |
| **Manje od 0:** | Vaš problem nije u tome što nemate znanja (to bi se dalo naučiti), već što vas ni sreća neće. A tu se malo šta može učiniti |

| | | | |
|---|---|---|---|
| 1 | E | 21 | C |
| 2 | C | 22 | B |
| 3 | B | 23 | A |
| 4 | D | 24 | C |
| 5 | E | 25 | A |
| 6 | B | 26 | C |
| 7 | D | 27 | A |
| 8 | D | 28 | D |
| 9 | D | 29 | C |
| 10 | C | 30 | C |
| 11 | B | 31 | D |
| 12 | A | 32 | A |
| 13 | B | 33 | E |
| 14 | B | 34 | B |
| 15 | B | 35 | C |
| 16 | B | 36 | D |
| 17 | B | 37 | D |
| 18 | B | 38 | D |
| 19 | D | 39 | A |
| 20 | E | 40 | D |

TAMANITE VIRUSE

NASTAVAK SA STRANE 53

### ...I DIZANJE SISTEMA

*Boot* sektor koji će biti učitan i predata mu kontrola određuje se identično kao i na zdravom disku. Upravo na tom mestu se nalazi prvi deo virusnog koda koji sada može slobodno da krade memoriju „ispred nosa" DOS-u. Početni deo *boot* sektora zaraženog računara (napomena: *boot* sektor se uvek učitava na apsolutnu adresu 0000:7C00) izgleda kao na listingu 1.

**Listing 1**

```
0000:7C00 EB1C     JMP    7C1E
0000:7C1E 33C0     XOR    AX,AX
0000:7C20 8ED0     MOV    SS,AX
0000:7C22 BC007C   MOV    SP,7C00
0000:7C25 8ED8     MOV    DS,AX
0000:7C27 A11304   MOV    AX,[0413]
0000:7C2A 2D0200   SUB    AX,0002
0000:7C2D A31304   MOV    [0413],AX
```

**Početni deo *boot* sektora zaraženog računara**

**Listing 2**

```
0000:7C30 B104     MOV    CL,04
0000:7C32 D3E0     SHL    AX,CL
0000:7C34 2DC007   SUB    AX,07C0
0000:7C37 8EC0     MOV    ES,AX
0000:7C39 BE007C   MOV    SI,7C00
0000:7C3C 8BFE     MOV    DI,SI
0000:7C3E B90001   MOV    CX,0100
0000:7C41 F3       REPZ
0000:7C42 A5       MOVSW
0000:7C43 8EC8     MOV    CS,AX
```

**Virusni kod**

Kao što se vidi, prve četiri instrukcije postavljaju vrednosti segmentnih registara DS i SS na nulu i registra SP na 7C00h. Na adresi 0000:0413h nalaze se dva bajta (word) koji govore koliko (u kilobajtima) ima memorije dostupne DOS-u. Sledeće tri instrukcije jednostavno umanjuju ovu vrednost za dva i time rezervišu dva kilobajta memorije za potrebe virusa. Nakon toga, virusni kod prebacuje sam sebe na početak upravo rezervisanog prostora u samom vrhu memorije i zatim nastavlja da se izvršava na toj, novoj adresi (listing 2).

U ovom delu koda zanimljiv je način na koji se izračunava adresa na koju se virusni kod prebacuje. Naime, na ulasku u ovu rutinu AX registar sadrži broj kilobajta u sistemu umanjen za dva (toliko je potrebno virusu). Ovaj broj se pretvara u segmentni deo adrese tako što se pomnoži sa 1024 i podeli sa 16. Kako je 1024/16=64, sledi da je u stvari potrebno samo ovaj broj pomnožiti sa 64, odnosno šiftovati ga za 6 mesta ulevo.

Nakon ovoga, virus će se pobrinuti za još nekoliko ne naročito zanimljivih stvari: učitće nastavak virusa (prvi sektor klastera koji je virus označio kao loš) na adresu XXXX:7E00h – tačno iza prvog dela, zatim učitaće originalni *boot* sektor na adresu 0000:7C00h, izmeniti interapt vektor 13h (koji će da se brine o budućim zarazama zdravih diskova) i, na kraju, startovaće program koji se nalazi u originalnom *boot* sektoru. Sve što dalje sledi za DOS je potpuno normalno, jedino je vaš sistem siromašniji za 2 KB.

Pitanje koje bi skeptici na kraju ovog izlaganja rado postavili glasi: „A zašto bi sve to nas interesovalo? Dobili smo *anti virus* i sve ostalo nas ne zanima." Uporni međutim dolaze do zaključka da su *virusi* u stvari delo Velikih Programera. Koliko puta ste se zapitali da li ste u stanju da napišete makar i najjednostavniji virus? Zbog svega toga budite sigurni da u svakom novom virusu kojeg „rasturite" ima mnogo toga čemu se treba diviti (najčešće zbog kompaktnosti koda, ali neretko i zbog njegove brzine).

Program koji sledi (listing 3.) nije potrebno posebno objašnjavati iz prostog razloga što ćete detaljne komentare naći u samom programu. Možda će vas uplašiti činjenica da ovaj program prepravlja FAT tabelu i menja *boot* sektor, ali za strah nema nema nikakvih razloga ukoliko ste sigurni da ste ga dobro prekucali. Program je testiran na desetak računara i **nigde** se nije povodom ovoga pojavio ni najmanji problem.

ARITMETIČKI IZRAZ

NASTAVAK SA STRANE 57

ostaje na aritmetičkom steku iznad šestice, a mi čitamo izraz dalje.

Pre svega, treba proveriti da li je podizraz završen zatvorenom zagradom, i prijaviti grešku ako nije. Pošto je sve u redu, prelazimo iza zatvorene zagrade i tamo nalazimo kraj niza.

Skidamo sa steka operaciju koren, izvršavamo je, skidamo operaciju sabiranja, izvršavamo i nju, skidamo „operaciju" početka niza i završavamo posao. Na aritmetičkom steku ostaje rezultat kome se nema šta zameriti, a stek operacija je ispražnjen.

### ALGORITAM I PROGRAM

Iz prethodnog primera postaje nam jasno na koji način se aritmetički izraz može računati, čak i ako čitamo elemente redom. Rešenje je prosto u korišćenju dva zasebna steka (jedan za operande, a drugi za operacije), pri čemu se operandi i operacije čuvaju sve dok se ne naiđe na operaciju nižeg prioriteta. Kompletan algoritam dat je na slici 1. Operacije smo uslovno označili sa *new* i *old*, pri čemu je *old* uvek ona operacija koja je na vrhu steka (stara operacija), dok je *new* ona koju smo upravo pročitali iz izraza (nova operacija). Kada poredimo operacije, poredimo ustvari njihove prioritete.

Program EVALUATE, koji je dat u okviru priloženih listinga, radi upravo po opisanom algoritmu. Kao stek operacija koristi se prosto običan mašinski stek mikroprocesora, jer su dva bajta (koliko inače staje na stek u jednoj PUSH naredbi) sasvim dovoljna da opišu svaku operaciju (niži bajt predstavlja kod operacije, a viši označava prioritet). Sami prioriteti su predstavljeni brojčanim vrednostima od jedinice do petice, na osnovu tabele OPTAB (prioritet funkcija definisan je u okviru potprograma FUNCTION, a prioritet negativnog predznaka u proceduri UNARMNS). Podizrazi u zagradama vrednuju se jednostavnim *rekurzivnim* pozivom programa EVALUATE. Rekurzija je moguća, pošto se svi parametri od značaja (operandi i operacije) odlažu tokom rada na stek.

**računari: AKCIJE • MARKETING • PROPAGANDA • PUBLICITET**

## MICRO COMPUTING

elektronika – hardware – software

ATARI® | AMIGA | IBM® PC/XT/AT

**Servis i popravci za preko 200 rezervnih dijelova**

ATARI ST 260/520/1040/Mega/TT
AT-SPEED, PC-SPEED
Hard diskovi (20—105 Mb)
Floppy diskovi 3,5" i 5,25"
Proširenje mem. 1 Mb, 2,5 Mb, 4 Mb
Modulator
Videodigitalizator
Pregradnja SF 354 u dvostrani
SM 124
SCART, CENTRONICS, KOMPOZITNI
T O S 1,4
Super miš

AMIGA 500/1000/2500/3000
PC-BOARD
Hard diskovi (20—60)
Floppy diskovi 3,5" i 5,25"
Pro. mem. 512 Kb, 1 Mb, 1,5 Mb, 2 Mb
Modulator
Videodigitalizator
Midi priključak
Big Agnus 8372A
SCART kabel
Sound-sampler
Super miš

### AT — SPEED i PC — SPEED

Poduzeća, fakulteti, instituti, škole, privatnici — pretvorite svoj ATARI ST u svestraniji računar. NORTON 6, 7, 6 grafičkih kartica, podržan kompletan hardware, CPU I 80286, 704 Kb+3 Mb EMS, Protected Mode, Software kao ACC sa promjenjivim fontovima, radi MS-WINDOWS 3 0 i dr.

**MICRO COMPUTING**

| | | |
|---|---|---|
| Fočanska 35<br>41000 Zagreb<br>Yugoslavia | Zagrebačka banka<br>30101 620-16<br>račun br 233546194 | tel 041/259-686<br>041/511-139<br>042/817-596 |

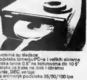
**Prosirite mogucnosti vaseg personalnog kompjutara, lokalne mreze i velikih kompjuterskih sistema. Nabavite magnetne jedinice cuvene firme**

## CIPHER

U proizvodnom programu firme su sledeci modeli:

**M880, M880/M881, M990GCR**

svi bazirani na poznatoj **CacheTape** mehanici.



Mogucnosti sistema su sledece:
- Razmena podataka izmedju PC-a i velikih sistema
- Koriscenje traka sirine 0.5" na koturovima do 10.5"
- Prenos podataka sa traka na disk i obratno
-PC-DOS Xenix, DEC verzija
- Velika brzina snimanja podataka 25/50/100 ips
- Nacin snimanja
9 track ANSI i IBM kompatibilni
**Phase Encode(PE) Group Code Recording(GCR)**
- Gustina zapisa 1600/3200/6250 BPI(pri 3200BPI sa trakom 2400ft daje 92MB)

Svaki sistem obuhvata sledece komponente samu magnetnu jedinicu interface plocu (SCSI AT Bus ili MCA) i software

**Sistemi neprekidnog napajanja strujom (UPS) firme:**

## EMERSON

snaga od **300VA** do **1.4KVA**



**i firme:**

## TOSHIBA

snaga od **1** do **15KVA**
Nemojte dozvoliti da vam propadne dugotrajni rad na kompjuteru ili na akviziciji podataka zbog banalnog nestanka struje u mrezi. Produzite vek vaseg skupocenog elektronskog ljubimca. Vec danas narucite UPS, a on ce vam se visestruko isplatiti pruzajuci zastitu od
- promenljivog mreznog napona
- strujnih udara
- gubitka napajanja iz mreze
Oblast primene UPS-a
- kod kompjutera i test opreme
- kod medicinske opreme i laboratorijskih instrumenata
- kod TV i audio sistema
- kod svih elektricnih uredjaja koji zahtevaju konstantan napon
**Isporuka odmah**

### Specijalna ponuda!

Pretvorite svoj izvanredni **HP LaserJet Series II** printer u vrhunski **PostScript** printer ugradnjom adaptera

**QMS JetScript 10900 din**

koga i sam **Hewlett Packard** preporucuje za svoj printer kao 100% **Adobe** kompatibilan. Povecajte brzinu **PostScript** stampe za preko 7 puta. Za kolicine vece od 3 komada samo **9900 din**
Cene su bez poreza na promet pa je neophodno dostaviti narudzbenicu sa izjavom
Za sva uredjaje dajemo garanciju od 12 meseci. Servis obezbedjen. Informacije, prodaja i demonstracija
**"LogicData"** (011)661-450

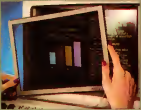
# SUNFLEX – SAD

PERI HARD ® YUGOSLAVIA

"PERIHARD" — poduzeće zadovoljnih korisnika!
ekskluzivni distributer za Jugoslaviju
Garešnička 36, pp 5030, 41040 Zagreb
tel 041/263-696, fax 041/263-714

Radite li više sati dnevno za monitorom ili terminalom?

## Vama je potreban SUNFLEX zaštitni filter za monitor!



PROTECTOR™

- jednostavna montaža na sve vrste monitora i terminala
- odvodi statički elektricitet i niskofrekventno zračenje katodne cijevi
- eliminira refleksije od ekrana
- čuva Vaše zdravlje
- smanjuje umor i naprezanje očiju
- povećava koncentraciju operatera, smanjuje postotak grešaka u radu, a time i produktivnost rada
- PERIHARD nudi izbor između filtera u tehnologiji mikrovlakana i staklenih filtera

VOLTFREE™

- instaliranje SUNFLEX Voltfree™ filtera direktno na katodnu cijev koje izvodi naš tehničar
- ovaj fleksibilni filter može se montirati na više od 3200 monitora u upotrebi u nas i u svijetu
- Voltfree™ filter je patentirani proizvod firme SUNFLEX – SAD
- SUNFLEX filtere je testirao najveći broj svjetskih instituta i potvrdio njihova korisna svojstva
- jedino PERIHARD ima optimalno rješenje za svaki monitor na YU-tržištu



NOVO

**Ergo Arm™** — **pomični, podešavajući oslonac za laktove**
**FootStool™** — **antistatički, podešavajući oslonac za stopala**

Beograd 011/172-413, 695-769
Ljubljana 061/374 380, 320-029
Sarajevo 071/219-175
fax 217-087
Novi Sad 021/332-449
Kranj 064/36-785, 34-039
fax 35-879
Doboj 074/41-535
Titograd 081/24-149
fax 43-173

**Dođite na "INTERBIRO '90" paviljon 8 , štand 8 , na akcionu prodaju – veleprodajne cijene!**




# BORLAND

## Novo!

### PARADOX 3.5:

- relacione baza podataka
- izvanredno jednostavna upotreba
- QBE (Query by ExaMPLE) — pristup bazi bez programiranja
- relacione naredbe join, outer join
- automatske podrške višekorisničkoj okolini
- zaključivanje zapisa, automatsko obnavljanje ekrana
- PAL — strukturirani jezik s debuggerom i editorom
- VROOMM tehnologija

## ZASTUPNIČKI PROGRAM:

**PROGRAMSKI JEZICI I ALAT:**

- Turbo Pascal 5.5
- Turbo Pascal 5.5 prof
- **Turbo C++**
- **Turbo C++Prof**
- Turbo Assembler/Debuger/ /Profiler

**POSLOVNI PROGRAMI**

- Reflex 2.0
- Turbo Lighting
- Sidekick 1.58
- Sidekick +
- **Superkey +**
- **Editor tekstova SPRINT**

- PARADOX ENGINE — veza relacione baze podataka PARADOX sa TURBO programskim jezicima. U pripremi verzija za rad sa MS WINDOWS

**Stotine hiljada korisnika programa za tabelarne kalkulacije prešle su na rad sa**

## QUATTRO® PRO

**Dobio je sva značajna priznanja kao najbolji program za tabelarne kalkulacije**

**Technical Excellence Award** 1989—PC Magazine
**Spreadsheet Product of the Year** 1989 InfoWorld
**Software Product of the Year** 1989 InfoWorld
**Top-rated High End PC Spreadsheet Rivew** 1990 InfoWorld
**Most Valuable Product (MVP)** 1989 PC Computing
**Award of Distinction** 1989 BYTE
**Best Buy & Top-Rated Spreadsheet** 1990 PC World
**1989 Software Product of the Year** PC User (UK)
**High-End Spreadsheet Editor s Choice** 1990 PC Magazine
**Best Numeric Business Application Award** 1989 Software Publishers Association
**Top-rated Consolidation Spreadsheet Review** 1990 PC Week
**Readers VIP Award, Spreadsheet Category** 1990 BYTE (tied for first)

**INFO WORLD u uporednom testu sa Excelom 2.1 i Lotusom 1-2-3 2.2 i 3.0 QUATRO PRO 1.0 bio je izabran za najbolji produkt ove vrste**

**NOVO: besplatan dodatak YU-znaci i meniji za QUATTRO PRO!**

Zahtevajte propagandni materijal i otkrijte sve o proizvodu koga se LOTUS tako boji!

### ZA OSTALE LICENCNE PAKETE POZOVITE NAS!

Pošto smo generalni jugoslovenski zastupnik za **BORLAND**, imamo sve pakete na raspolaganju! Isporuka odmah! Za BORLAND preuzimamo takođe registraciju kupaca, njihovo obaveštavanje o izlasku novih verzija i jeftinu zamenu starih (upgrade)! Školskim ustanovama nudimo školske verzije svih paketa po znatno nižim cenama! Za našu ponudu mašinske i mrežne opreme (Novell) molimo vas da nas pozovete!

**GENERALNI I EKSKLUZIVNI ZASTUPNIK ZA JUGOSLAVIJU**

MARAND

Intenarng d.o.o.
61000 Ljubljana
Kardeljeva ploščad 24
Tel: (061) 371-114
Fax: (061) 342 757

## OPTIČKI DISKOVI



Ten X Technology, Inc.

| Model | Sistem | Izmenljivi Disk |
|---|---|---|
| **Opti Xchange** 650 MB Multi-Funkcijski Optički Podsistem (5 25") | $ 7,195 | $ 300 |
| **OptiWin 650** 650 MB WORM optički podsistem (5.25") | $ 5,395 | $ 175 |
| **OptiWin 940** 940 MB WORM Optički Podsistem (5 25") | $ 6,395 | $ 180 |
| **OptiWin 2400** 2.4 GB WORM Optički Podsistem (12") | $ 20,735 | $ 720 |
| **OptiWin 6400** 6.4 GB WORM Optički Podsistem (12") | $ 36,065 | $ 720 |

- Jedinstveni **OCU** (Optical Conversion Unit) za sve modele omogućava njihovo priključenje na bilo koji računar sa SCSI kontrolerom
- **OCU** vrši on-line komprimovanje i dekomprimovanje zapisa
- Nisu potrebni nikakvi drajveri i softver za upravljanje datotekama
- Kompatibilnost sa 286, 386, Mac, DEC, SUN, TI, IBM PS/2 i RT, ...
- Nezavisnost od operativnih sistema DOS, UNIX, XENIX, VMS, ULTRIX, ...
- **Opti Xchange model podržava i "Piši-Briši" i WORM diskove**

Plaćanje u dinarima + cca 30% dažbine

## CAD SISTEMI - RADNE STANICE - PC

**INTERBIRO '90 - prezentacije na štandu El Bull HN**

Nth GRAPHICS



**Nth Engine 550**

- Transpjuterski CAD kontroler za profesionalni rad sa AutoCAD, VersaCAD, CV Personal Designer, ...
- Zoom 0 2 sec
- Ptičiji pogled, Zoomslide, Template
- HYDRA
  - Senčenje AutoCAD i VersaCAD modela za 3 sec
  - Animacija i Walk-Thru u realnom vremenu
- Nth TV, video opcija za prezentiranje projekata

**Nth Engine serije 300**

**Nth Drive softver za AutoCAD 386**

- Nth Display List mogućnosti i funkcije
- Nth Zoom, Nth Ptičiji pogled, Zoomslide, Template

**PC 286, 386, 486 Bull HN, COMPAQ, Austin Computers, Kompatibilni Monitori, Ploteri, Skeneri, Digitajzeri**
**COADE Software CAESAR II i ProVessel**

## PROFESIONALNA GRAFIKA I ANIMACIJA

AT&T

TRUEVISION®

**AT&T GSL Softver**

TOPAS, RIO, TIPS, Panorama, Logo Editor, ..

**Truevision**

Targa, ATVista i NuVista kontroleri

**AV BV Graphics** *Andrićev Venac 4, 11000 Beograd, Tel. (011) 345-277, Fax. (011) 345-277, 336-713*

*U Sloveniji: CGS, Cesta V Podborši 2 Tel 061/373 777 Fax 061/552-140*

# MALI OGLASI

**Običan mali oglas do 10 reči stoje 160 dinara, a svaka dalja reč 16 dinara. Centimetar uokvirenog oglasa staje 200 dinara, s tim što se mogu zakupiti najmanje 2 cm i što oglas ne može biti viši od 5 cm.**

**Postupak za objavljivanje vašeg malog oglasa je jednostavan: pošaljite redakciji tekst oglasa zajedno sa foto-kopijom uplatnice kojom ste na žiro-račun RO BIGZ 60802-603-23264 (za Male oglase u „Računarima") uplatili odgovarajući iznos.**

**Rok za slanje: do 10. u mesecu za naredni mesec.**

NA PRODAJU AMSTRAD PC 1512 D/D FLOPPY 5 3/4' KOLOR MONITOR AMSTRAD PRINTER DMP 3000 SA SOFTWERSKIM PAKETOM CENA 20 000 DIN. MOGUĆ DOGOVOR TEL 011/332-441 RADNIM DANOM OD 8 00 OD 12 00

ATARI ST PROGRAMI I IGRE BESPLATAN KATALOG NAJPOVOLJNIJE CENE SPASOVSKI DRAGAN MARŠALA TITA 15/19, 91220 TETOVO

Ugrađujem YU-karaktere u SVE vrste video-kartica (hercules/CGA/EGA/VGA i kombinovane) po NOVOM IBM standardu (CP852) po starim YU i specijalnim standardima. Izrađujem drajvere ... štampače Tel 011/592-087

Srpska ćirilica i YU latinica
Ugradnja u Epson LQ i druge štampače grafičke kartice (Hercules CGA EGA VGA ) Tel 011/347 509
403-205

Veoma povoljno prodajem originalne koprocesore 80287 8 Intel Mašić Nermin 071/528-629

COMPUTER SERVICE
VIII Vrbik 33a/6 41000 Zagreb
tel 041/539 277 719 892
od 10 do 12 i od 15 do 17 sati
Stranke primamo od 10 do 12 sati
- SPECTRUM COMMODORE ATARI AMSTRAD
- brzi i kvalitetni popravci
- memorijska proširenja
- prodaja kompjutera flopy drive interface a joystick a
- kablovi za povezivanje sa printerom monitorom televizorom
- ZX centronic interface joystick interface
- C64 eprom modul centronic kabel
- rezervni dijelovi za kompjutere
Tražite besplatni katalog

KOMPJUTEROM DO ZARADE posao u Vašem stanu za besplatne informacije pošaljite adresirani koverat sa markom Stojković Nenad 21000 Novi Sad Put partizanskih baza 8

| DISKETE | |
|---|---|
| 5,25" (360 Kb) | 10 din. |
| 3.5" (720 Kb) | 21 din. |
| 5,25" (1.2 Mb) | 21 din. |

071/ 214-319, 628-519 ili 455-640

**PC SHOP**
- računari, štampači, periferije
- diskete, kablovi, servis

**Ponuda meseca**
- Modemi ATI (Canada), sa ili bez MNP opcije, 1200 do 9600 bps izuzetan kvalitet, reklamne cene
- Računar 386-25MHz, CACHE, SUPER VGA color 1024*768, samo... 38 500 d

Milene Kojiće 24
14220 Lazarevac 011/8124-686

80386—25 MHz 4 Mb RAM-a hard 120 Mb NEC 3D monitor VGA kartica Tel 034/63-011

Prodajem kazete za COMODORE 64 Mušur i Neda Bu...nova ... 58000 Split tel 058/36 ...71

FAX kartica (prijem, slanje) za XT/AT/386 Nova kvalitetna more se prodati. Trake za sve vrste štampača povoljno osvežavam tečnošću iz USA. Tel 011/434-480

Sve za vašeg mezimca. Diskete, strimeri koprocesori, riboni 079/788-166

Diskete superkvalitetne 5 25' i 3 5' prodajem Preduzećima i školama šaljem račun. Tel: 055/63-091

Preferans PC XT/AT originalni zaštićeni softver na srpskohrvatskom + uputstvo šaljem pouzećem (200 din ) Tel 018/715-844

Prodajem Atari 520 ST +, 1 MB RAM, NEC 1037A flopi SM124 monitor, diskete Nenad 011/673 733

## NAJVEĆA RAČUNARSKA BERZA U JUGOSLAVIJI

**U SVAKOM BROJU ČASOPISA „RAČUNARI"**
**VELIKI KOMERCIJALNO-INFORMATIVNI BLOK**

**akcije • marketing • publicitet • propaganda**

plus

**NON-STOP 24 SATA DNEVNO**
**NA SISTEMU ZA MODEMSKE KOMUNIKACIJE**

**sezam commerce**
(opširnije na str 67)

BRZO JEDNOSTAVNO JEFTINO

## PRODAJA LICENČNE PROGRAMSKE OPREME PO POŠTI

**TRIAS WTC** COMPUTER DIVISION, Dalmatinova 4, LJUBLJANA

| BORLAND | |
|---|---|
| EUREKA | 2340 |
| Paradox 3.5 | 10840 |
| Paradox 386 | 12100 |
| Paradox Network | 14040 |
| Quattro Pro | 4900 |
| Reflex 2.0 | 3600 |
| Sidekick Plus | 3600 |
| Turbo Basic DB Tbls | 1400 |
| Turbo C 2.0 | 2100 |
| Turbo C Professional | 3500 |
| Turbo C ++ | 2900 |
| Turbo C ++ Professional | 3900 |
| Turbo Pascal Graphix Tools | 1400 |
| Turbo Pascal 5.5 | 2100 |
| Turbo Pascal Pro | 3430 |
| Turbo Prolog | 2160 |
| Turbo Prolog Toolbox | 1400 |

| CAD | |
|---|---|
| Acad 10 | pozovite |
| Autosketch 2.0 | 2050 |
| Design CAD | 3450 |
| Design CAD 3 D | 4510 |
| Draft for Windows | 9700 |
| Draft Ultra | 5610 |
| Generic CADD level 3 | 3670 |
| Math CADD 2.5 | 6480 |
| Mont Generic Symbol | 650 |
| 3 Drelling | 3130 |

| CASE TOOLS | |
|---|---|
| Erase | 7100 |
| Meta Design | 5400 |
| Clarion Prof Developer | 14100 |
| dAcalyst | 4500 |
| Pro-C | 6900 |

| COMMUNICATIONS | |
|---|---|
| Blast PC | 3670 |
| Crosstalk Mark IV | 3130 |
| Crosstalk for Windows | 2710 |
| Crosstalk XVI | 2290 |
| Remote 2 | 2390 |
| Smartcom II | 1940 |

| DATABASE | |
|---|---|
| Ask Sam | 3720 |
| Clipper | pozovite |
| Database V 4.0 | 10800 |
| Data Perfect | 6690 |
| Dbase IV | 10300 |
| DBXL Diamond | 3240 |
| Enable OA | 9720 |
| Foxbase Plus | 4100 |
| Foxbase Plus Multi | 6690 |
| Foxbase Plus M/U runtime | 7100 |
| Foxbase+Runtime S.U | 5770 |
| Foxbase Plus 386 | 6690 |
| FOX Pro | 10150 |
| FOX Pro LAN | 14250 |
| FOX Pro Runtime | 6870 |
| Framework III | 9600 |
| Q&A | 4960 |
| Quicksilver Diamond | 7560 |
| Rapidfile | 4750 |
| Rbase DOS | 10580 |
| Rbase OS/2 | 13200 |

| | |
|---|---|
| Superbase 4 (Win) | 8400 |
| Symphony | 9120 |

| DESKTOP PUBLISHING | |
|---|---|
| Bitstream 1 family | 1900 |
| Byline | 4210 |
| Gem Desktop Pub. | 3700 |
| Impress | 2700 |
| Plotter in a Cartridge | 4300 |
| Print a Plot | 2450 |
| Laser Control Card | 8340 |
| Pacific Page Postscript | 3420 |
| Freedom of Press | 3900 |
| Flowcharter(Win) | pozovite |
| Pacific Page | 9930 |
| 25at One | 5610 |
| 35at One Yuch | pozovite |
| Pagemaker OEM III | 3000 |
| Ventura 3.0 /Gold | 10400 |

| | |
|---|---|
| Diagraph 2000 | 5400 |
| Freelance Plus | 3070 |
| Gem Artline | 3600 |
| Gem Draw Plus | 3600 |
| Gem Graph | 3600 |
| Gem Presentation Team | 3600 |
| Graphwriter | 7540 |
| Harvard Graphics | 6400 |
| Halo | 4800 |
| Mapmaster | 5720 |
| Master Graphics | 6600 |
| Meta Window + | 3720 |
| Object Graphics | 5400 |
| PC Paintbrush IV+ | 2550 |
| The Animator | 5400 |
| Windows Draw Plus | 6900 |
| Windows Graph Plus | 7260 |

### HOT TEN OCTOBER

1. MS WINDOWS 3.0
2. VENTURA 3.0 /GOLD
3. PCTOOLS DELUXE 6.0
4. NORTON ADVANCED 5.0
5. DESIGN CAD
6. TURBO C++ PROF
7. QEMM - 386 5.0
8. WORDSTAR 6.0
9. PROCOMM PLUS 1.1B
10. PC PAINTBRUSH IV +

| EXPERT SYSTEMS TOOLS | |
|---|---|
| 1st CLASS Fusion | 19500 |
| Business Win | pozovite |
| LEVELS Object | pozovite |
| KnowledgePro | 5450 |
| KnowledgePro(Win) | 12600 |
| NeuralWorks Prof I | 8400 |
| NeuralWorks Prof II | 18000 |
| Smalltalk V286 | 3020 |
| Smalltalk V/PM | 6480 |
| Smalltalk 80 | pozovite |
| Super Expert | 3100 |
| Rulemaster 3 | 10200 |

| GRAPHICS | |
|---|---|
| ADOBE Illustrator(Win) | 6480 |
| ARTS & LETTERS Composer | 5840 |
| ARTS & LETTERS Graphics Editor | 9720 |
| Calendar Creator + | 1860 |
| Colorix | 2370 |
| Corel Draw | 7340 |
| Designer | 9720 |
| Diagrammaster | 4750 |
| Mels II Draw | pozovite |

| OBJECT PROGRAMMING | |
|---|---|
| Actor | 10500 |
| Complete C | 8900 |
| Eiffel Devel Kit | 14100 |
| Eiffel Ext Envir | 18400 |
| Zortech C++ | 5100 |

| SCIENCE | |
|---|---|
| Chi Writer | 2100 |
| Dadisp | 10800 |
| Circad PCB | 18900 |
| Statgraphics | 10450 |
| T | 7200 |
| PC Globe 3.0 | 840 |
| Professor DOS | 600 |

| SPECIAL LANGUAGES | |
|---|---|
| APL Plus | 9930 |
| Lattice C | 3670 |
| Modula 2(Logitech)Dev Sys. | 4100 |
| TopSpeed Modula 2 | 7390 |
| RM Cobol 85 (full) | 17280 |
| RM Fortran | 8420 |
| RM Cobol | 13170 |
| Power Basic | pozovite |

| MICROSOFT | |
|---|---|
| MS BASIC 7.1 | 6890 |
| MS C Compiler | 6120 |
| MS Chart | 4040 |
| MS Cobol 3.0 | 12200 |
| MS Excel | 6250 |
| MS Flight Simulator 4 | [illegible] |
| MS Fortran 5.0 | 6390 |
| MS Pascal | 3900 |
| MS Project | 6850 |
| MS Windows 3.0 | 2300 |
| MS Windows Dev Toolkit | 7340 |
| MS Word | 4900 |
| MS Word Window | 6100 |

| OPERATING SYSTEMS | |
|---|---|
| Concurrent DOS 386 | 7600 |
| Concurrent DOS XM | 6150 |
| MS DOS 4.01 | 2500 |
| PC Mos | 4400 |
| PC Mos 5-user | 14100 |
| DRDOS 5.0 | pozovite |

| PROJECT MANAGEMENT | |
|---|---|
| Harvard Project III | 9100 |
| Primavera | 1400 |
| Super Project Expert | 9720 |
| Super Project Plus | 5610 |
| Timeline 4.0 | 10800 |
| Project Scheduler 4 | 10470 |

| SPREADSHEETS | |
|---|---|
| LOTUS 123 2.2 | 6500 |
| LOTUS 123 3.0 | 7700 |
| Plan Perfect | 5610 |
| Supercalc V | 7910 |
| VP Planner 3-D | 3450 |
| Wingz | pozovite |

| UTILITIES | |
|---|---|
| inFocus 1.01 | pozovite |
| Norton Advanced 5.0 | 1790 |
| Norton Commander 3.0 | 1700 |
| Office LAN 6.0 | 6040 |
| PC Tools 6.0 | 1850 |
| UI Programmer 2.0 | 7560 |
| Watchdog | 3990 |
| Relay Gold 4.0 | 3580 |
| Laplink III | 2400 |
| Disk Technician Adv 6.0 | 3800 |
| Deskview | pozovite |
| Brooklyn Bridge | pozovite |
| Site Lock | pozovite |
| Turnstyle | pozovite |
| Fastback Plus 2.09 | 2700 |
| Desk Link 2.21 | 2850 |
| Turbo EMS | 1200 |
| Magellan | 2450 |
| Mace 1990 | 1720 |
| Puzzas Plus | 1720 |
| Qemm 386 | 1200 |
| Qram | 960 |

| WORD PROCESSORS | |
|---|---|
| AMI Professional | 6350 |
| Multimate V 4.0 | 6370 |
| Notabene | 5830 |
| Q&A Write | 3240 |
| WordPerfect 5.1 | 6610 |
| Wordstar 6.0 | 3950 |

Nudimo i svu dodatnu opremu za laserske štampače: POSTSCRIPT (SW + HW), MEMORY UPGRADE, FONT CARTRIDGES... Posredujemo ... kod nabavke ... (skenera), OCR programa, grafičkih stanica i ostale opreme. Generiramo Yu fonte. Organiziramo ON-SITE seminare za posamezne produkte. Dobavljamo i ... programe.

Dobava 10-14 dana posle narudžbe po pošti na željenu adresu.

ŽIRO RAČUN: 50100-601-62207 pri SDK Ljubljana

TELEFON : 041 316-343 , 310-033 (od 9h do 17h)

A N Y S O F T W A R E A N Y Q U E S T I O N A N Y T I M E

# elder computers

I RAM-G d.o.o. LJUBLJANA, Kumrovška 7 tel (061) 346-492 omogućsveju svojim kupcima BESPLATAN dvodnevni seminer iz osnova računara PC u svom obrazovnom centru u Ljubljeni, Pod gozdom 10, tel (061) 327-770 i (061) 346-492 Dodatne povoljnosti pri kupovini računara u septembru i oktobru.

## PONUDA 286 16MHz

| | |
|---|---|
| Osnovna ploča 80286 16 MHz | 388 DEM |
| Memorija RAM 1 Mb | 324 DEM |
| Kućište BABY AT + nasp 200 W | 196 DEM |
| Video Kartica AUTODUAL | 83 DEM |
| Kontroler FDD - HDD | 160 DEM |
| Floppy disk drive 1 2 MB | 173 DEM |
| Hard disk | |
| SEAGATE ST 252 40 Mb | 672 DEM |
| Dodatne kartica MULTI I/O | 44 DEM |
| Tastatura 101 tipke tip CHERRY | 123 DEM |
| Monokromatski MONITOR 12" PW | 186 DEM |
| Printer SEIKOSHA SP 180 | 362 DEM |
| Kabel CENTRONICS za printer | 13 DEM |
| UKUPNO | 2 733 DEM |
| **PONUDA** | **2 800 DEM** |

## OSNOVNE PLOČE

| | |
|---|---|
| XT 4 77 - 12 MHz | 126 DEM |
| AT 286 12 MHz | 321 DEM |
| AT 286 16 MHz | 386 DEM |
| AT 286 NEAT 16 MHz | 585 DEM |
| AT 286 NEAT 20 MHz | 650 DEM |
| AT 285 NEAT 25 MHz | 800 DEM |
| 80386 SX 16 MHz | 838 DEM |
| 80386 SX 20 MHz | 822 DEM |
| 80386 25 MHz | 1 820 DEM |
| 80386 25 MHz CACHE | 2 383 DEM |
| 80386 33 MHz CACHE | 3 080 DEM |
| 80486 25 MHz | 8 820 DEM |

## PONUDA 386 SX-16 MHz

| | |
|---|---|
| Osnovne ploča 80388 SX 16 MHz | 838 DEM |
| Memorija RAM 1 MB | 324 DEM |
| Kućište KACER SX + nap 200 W | 345 DEM |
| Video kartica VGA 800 × 600 | 194 DEM |
| Kontroler FDD - HDD interleavel 1 | 192 DEM |
| Floppy disk drive 1 2 Mb | 173 DEM |
| Hard disk | |
| SEAGATE ST 252-40 Mb | 872 DEM |
| Dodatna kartica MULTI I/O | 44 DEM |
| Tastatura 101 I tip CHERRY | 123 DEM |
| Monokr MONITOR VGA 1024 × 768 | 320 DEM |
| Printer SEIKOSHA SP 2000 | 560 DEM |
| Kabel CENTRONICS za printer | 13 DEM |
| MIŠ | 93 DEM |
| UKUPNO | 3 891 DEM |
| **PONUDA** | **3 800 DEM** |

## KUĆIŠTA

| | |
|---|---|
| CASE BABY AT & POWER 200 W | 195 DEM |
| CASE ELEGANT & POWER 200 W | 260 DEM |
| CASE KACER SX & POWER 200 W | 345 DEM |
| CASE MINITOWER & POWER 200 W | 298 DEM |
| CASE MINIT & P 200 W + display | 320 DEM |
| CASE MIDITOWER & POWER 200 W | 385 DEM |
| CASE TOWER & POWER 230 W | 438 DEM |

## VIDEO KARTICE

| | |
|---|---|
| Monokrom HERCULES + Printer | 49 DEM |
| AUTODUAL + PRINTER | 83 DEM |
| SUPER EGA CARD | 185 DEM |
| VGA CARD 800 × 600 | 194 DEM |
| SUPER VGA CARD 1024 × 768 | 360 DEM |

## PONUDA 386 - 25 MHz

| | |
|---|---|
| Osnovna ploča 80386 25 MHz | 1 820 DEM |
| Memorija 2 Mb RAM | 648 DEM |
| Kućište TOWER + nap 230 W | 438 DEM |
| VIDEO kartica VGA 1024 × 768 | 360 DEM |
| Kontroler FDD - HDD interleave 1 1 | 192 DEM |
| Floppy disk drive 1 2 MB | 173 DEM |
| Floppy disk drive 1 44 MB | 185 DEM |
| Hard disk | |
| SEAGATE ST 4096 | 1 424 DEM |
| Dodatna kartica MULTI I/O | 44 DEM |
| Tastatura 101 tipke tip CHERRY | 123 DEM |
| Kolor MONITOR VGA MULTIS | 1 130 DEM |
| LASER print SEIKOSHA OP 105A | 3 270 DEM |
| Kabel CENTRONICS za printer | 19 DEM |
| MIŠ | 93 DEM |
| UKUPNO | 8 620 DEM |
| **PONUDA** | **8 300 DEM** |

## HARD DISK

| | |
|---|---|
| HDD SEAGATE ST 225 20 Mb | 430 DEM |
| HDD SEAGATE ST 124 20 Mb | 445 DEM |
| HDD SEAGATE ST 138 A -32 Mb | 560 DEM |
| HDD SEAGATE ST 251 1 40 Mb | 675 DEM |
| HDD SEAGATE ST 225 40 Mb | 672 DEM |
| HDD SEAGATE ST 157 - 45 Mb | 733 DEM |
| HDD FUJITSU S 340 - 40 Mb | 783 DEM |
| HDD SEAGATE ST 1096 - 84 Mb | 999 DEM |
| HDD CONNER BF 104 - 104 Mb | 1 424 DEM |
| HDD SEAG ST 1126A 111 Mb | 1 338 DEM |
| HDD SEAG ST 1238A 211 Mb | 2 325 DEM |
| HDD TOSHIBA CS 330 A-330 Mb | 4 235 DEM |

## MONITORI

| | |
|---|---|
| Monokrom MONITOR 12" zelen | 175 DEM |
| Monokrom MONITOR 12 PW | 196 DEM |
| Monokrom MONITOR 14" zelen | 220 DEM |
| Monokrom MONITOR 14" amber | 230 DEM |
| Monokrom MONITOR 14" PW | 240 DEM |
| Monokrom VGA 1024 × 768 14" PW | 320 DEM |
| Kolor EGA 640 × 350 14" | 780 DEM |
| Kolor VGA 800 × 600 14" | 825 DEM |
| Kolor VGA MULTISYNC 14" | 1 130 DEM |

## DODATNE KARTICE

| | |
|---|---|
| MULTI I/O CARD za AT | 44 DEM |
| MULTI I/O CARD za XT | 76 DEM |
| RS 232 CARD | 30 DEM |
| 2 × RS 232 CARD | 40 DEM |
| 4 × RS 232 CARD | 175 DEM |
| PRINTER CARD | 24 DEM |
| GAME CARD | 30 DEM |
| KLOCK CARD za XT | 30 DEM |
| RAM CARD 576 Kb za XT | 66 DEM |
| RAM CARD 2 5 Mb za AT | 148 DEM |
| RAM CARD 8 Mb za 386 | 150 DEM |

## FLOPPY DISK

| | | |
|---|---|---|
| FDD FUJITSU 5" 1/4 | 360 Kb | 145 DEM |
| FDD FUJITSU 5" 1/4 | 1 2 Mb | 173 DEM |
| FDD FUJITSU 3" 1/6 | 720 Kb | 165 DEM |
| FDD FUJITSU 3" 1/2 | 1 44Mb | 185 DEM |

## KONTROLERI

| | |
|---|---|
| KONTROLER za floppy disk za XT | 26 DEM |
| KONTROLER za floppy disk za AT | 80 DEM |
| KONTROLER za GARDE DISK za XT | 98 DEM |
| HDD FDD KONTROLER interl 1 2 | 160 DEM |
| HDD FDD KONTROLER interl 1 1 | 192 DEM |

## MEMORIJE RAM

| | |
|---|---|
| RAM 41256 10 | 7 DEM |
| RAM 41256 8 | 8 DEM |
| RAM 44256 10 | 28 DEM |
| RAM 44256 - 8 | 32 DEM |
| RAM 411000 10 | 24 DEM |
| RAM 411000 8 | 27 DEM |
| RAM MODULE SIM 256 K | 96 DEM |

Sve navedene cene su maloprodajne neto cene Za veće količine zatražite telefonom našu ponudu Celokupni ponuđeni materijal ima dvogodišnju garanciju Cene mogu da se menjaju u skladu sa kretanjem US dolara Za nabavku navedenog materijala obratite se direktno našem sedištu u Trstu Preporučujamo vam da pre kupovine telefonirate zbog rezervacije Naše radno vreme 9,00–12,00 i 15,00–19,00, osim subotom i nedeljom Računarsku opremu šaljemo i železnicom

**SERVIS RAM-G d o o LJUBLJANA**
Kumrovška 7 tel (061) 346 492
**PREDSTAVITVENI CENTER**
Pod gozdom 10 tel (061) 327 770
Radno vreme radnim danom 8–14, 15–18, subota 9–14

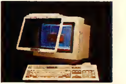
**SERVIS ELCOM KOPER**
KOPER JLA 8, tel (066) 24 977, 23-685
telefax (066) 24 881

ELDER COMPUTERS TRST, Ulica F Severo 8 (kod suda) Telefon 9939 40 382205 III 9939 40 382004 FAX 9939 40 382081
BANKE TRŽAŠKA KREDITNA SLOVENSKA BANKA C/C 4288/23 – BANCA DEL FRIULI C/C 1661 – IBI C/C 15305

# M U L T I G U A R D

## ČUVAJTE ZDRAVLJE : poboljšajte svoje uslove rada!

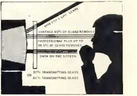
## Izbegnite neprijatne glavobolje i umor.



**Zaštitni filteri MULTIGUARD su pravo rešenje za vaše probleme. MULTIGUARD filteri, koje proizvodi poznata engleska firma OCLI, najnovije su dostignuće tehnike.**

**Filter je napravljen od šest međusobno slepljenih slojeva, a na prednjoj i zadnjoj strani filtera nalepljeno je specialno aluminijumsko kaljeno staklo, koje štiti filter od mehaničkih oštećenja, a istovremeno odlično odvodi statički elektricitet.**

IN SPACE OR THE OFFICE, HELPING PEOPLE TO SEE BETTER IS OCLI'S BUSINESS




**EKSKLUZIVNI PREDSTAVNIK ZA JUGOSLAVIJU:**

**Rožna dolina V/23**
**61000 LJUBLJANA**
**Tel.: (061) 261-300**

**MULTIGUARD ™ *filteri su sastavni deo stakala i ekrana na kontrolnim instrumentima vasionskog taksija SPACE SHUTTLE***


EPSON
VELIKO IME, POUZDAN KVALITET!
Epsonove proizvode između ostalih prodaju i
Avtotehna, Titova 36 i Celovška 228, Ljubljana
Mladinska knjiga Veletrgovina, Titova 145, Ljubljana
Mladinska knjiga Trgovina, Titova 3, Ljubljana
Gambit p.o., Titova 8, Ljubljana
ATR Ljubljana
Mogućnost leasing prodaje!
EPSON
EPSON
doo
CELOVŠKA 175 • YU
61107 LJUBLJANA
REPRO
LJUBLJANA
TELEFON 061/552-341, 552-150, 554-450 FAX (061) 552-563
TLX 31 639 yu-antena p.p. 69
POUZDANO IME,
VELIKI KVALITET!

Duško Savić

# OBJEKTNO PROGRAMIRANJE na Turbo Pascalu 5.5 – uvod –

Programeri već decenijama pokušavaju da odgovore na jednostavno pitanje: „Kako pisati dobre programe?" Pokazalo se da su dobri programi modularni, lako shvatljivi, da se mogu primenjivati i u situacijama za koje nisu specijalno pisani, da se mogu srazmerno lako menjati i dopunjavati. Sedamdesetih godina vodile su se žustre debate oko tzv. „struktuiranog programiranja" kao metode za pisanje programa, a epilog danas znamo: struktuirano programiranje ušlo je u kulturno bastinu svakog programera. Uporedo – mada bez pompe – evoluirale su i druge programske metodologije, među kojima je objektno programiranje najznačajnije.

Najveći broj računarskih jezika podržava model izračunavanja koji se može sažeti ovako: aktivne procedure deluju nad pasivnim podacima. Svi najpoznatiji programski jezici su u ovoj grupi – paskal, fortran, Algol, PL/1, kobol, bejzik itd. U njima postoji oštra razlika između podataka i informacije koje se obrađuju i procedura (potprograma, funkcija) koje ih obrađuju. Procedure i podaci su – što se tih jezika tiče – različiti entiteti, pa se programiranje svodi na konstruisanje programskih modula i njihovo pozivanje u toku izvršavanja programa. Svaki jezik na svoj način definiše prenošenje podataka u procedura kao i oblast važenja imena promenljive.

Objektni jezici zasnivaju se na ideji da jedan jedini entitet – objekat – treba da obuhvati i podatke i procedure. Podaci u objektu mogu biti promenjeni samo preko procedura i funkcija koje pripadaju tom objektu. Procedure i funkcije iz objekta jednim imenom nazivamo metodima. Poziv metode naziva se *slanjem poruke* objektu. Objekti odgovaraju na poruke tako što stvaraju nova objekte i vraćaju ih kao rezultate. Svaki objekat je poput nezavisnog procesora čije se ponašanje može doku čiti samo na osnovu odgovore koje odaje. Na primeru običnog sabiranja dva broja uvidećemo ogromnu razliku između ova dva pristupa.

Pretpostavimo da treba sabirati dva broja

4

U konvencionalnim programskim jezicima gornje izračunavanje bi se objasnilo na sledeći način: Naredba sabiranje + primenjuje se na dva celobrojne argumenta 2 i 4 i vraća rezultat 6. U pravom objektnom jeziku rezultat će takođe biti 6, ali način na koji se do njega dolazi je drugojačiji. Poruka „+" poslata je celobrojnom objektu 2 a argument je 4. Objekat koji prima poruku (ovde 2) samostalno odlučuje kako će se naredba izvršiti, dok u jezicima tipa paskala aritmetički operator + komanduje izvršenjem. Komunikacija sa objektima uvek je dvosmerna: rezultat koji je i sam objekat (u gornjem primeru celobrojni objekat 6) vraća se kada se poruka pošalje objektu.

Programiranje u pravom objektnom jeziku svodi se na svega dve operacije:

1. Prepoznavanje objekata i poruka u problemu o kojem želimo da pišemo program.

2. Određivanje odgovarajuće strukture podataka koje podatke interno predstavlja u okviru objekta, uz razvoj algoritama koji odgovaraju na poruku (tj. izvršavaju naredbu).

Objekat ima dva vida: spoljašnji i unutrašnji. Spoljašnji nas obaveštava šta dati objekat radi, a unutrašnji precizira *kako* se to postiže. *Ne postoji način da se promeni stanje objekta osim slanjem poruka.* Takva komunikacija je veoma prirodna i preuzeta je iz svakodnevnog života. O tome šta neka druga osoba misli može se suditi samo po porukama koje prima ili odaje, a svaki individuum zna svoje unutrašnje stanje i kako reaguje na primljene poruke. Prema većini objekata u realnom životu ponašamo se na isti način – slanjem i primanjem poruka, tako da je objektno programiranje prirodan nastavak uobičajenog ljudskog ponašanja. U stvari, u objektnim jezicima *svaki* program je emulacija nekog realnog procesa, i baš zato je programiranje na ovim jezicima najproduktivniji vid programiranja, takoreći – dečje igra. Ovaj termin nije slučajno odabran, jer prvi objektni jezik *Smalltalk* nastao je upravo iz želje da se napravi snažan računar za decu.

Sredinom sedamdesetih godina kompanija XEROX stvorila je eksperimentalni naučno razvojni računarski centar u gradiću Palo Alto u Kaliforniji. Skraćeno, centar se zvao PARC (Palo Alto Research Center), i učinio je modernom računarstvu neprocenjive usluge. XEROX je skoro decenijm plaćao vrhunske računarske stručnjake da bi se isključivo bavili razvojem i eksperimentisanjem – bez ikakve obaveze da nešto konkretno stvore i ponude tržištu. Posle desetak godina PARC je zatvoran, ali je ostavio za sobom takve „svakodnevne" stvari kao što su miševi, ikone, bit-po-bit definisani ekrani (*bit-mapped display*), meniji, preklapajući prozori, projekat *Dynabook* i prvi objektni programski jezik – *Smalltalk*.

*Alan Kay* (Alen Kej), jedan od istraživača u PARC-u, stvorio je viziju računara *Dynabook*, koji je trebalo da ima sledeće osobine: cenu od svega nekoliko stotina dolara, veličinu školske sveske, ogromnu memoriju, ravan ekran osetljiv na dodir, bit-po-bit programabilan grafički ekran, audio izlaze i komunikacione mogućnosti. *Smalltalk* je trebalo da obezbedi softversku podršku. Razvoj *Smalltalka* produžio se na skoro deset godina, a završene verzije obnarodovana je 1980-te godine pod imenom *Smalltalk-80*. Prva knjiga o *Smalltalku* izašla je tek 1983. godine. Od tada ovaj i drugi objektni jezici (*Actor*, *HyperTalk*) – kao i cela ideja objektnog programiranja – snažno dobijaju na zamahu.

*Smalltalk* je nastajao i kao jezik i kao operativni sistem, tako

da uvek sadrži kompletnu radnu okolinu: interpreter, editor teksta, prozore, naredbe za rad sa datotekama, komunikaciju sa mišem, *debugger*, grafiku visoke rezolucije, prevodilac, *linker*, menije, fontove, uvršćavanje mašinskih programa, uopšte — sve što je programeru potrebno za svakodnevan rad. Projekat *Dynabook* i sve što je iz njega proizišlo predstavljaju jedan od ključnih zaokreta u istoriji računarstva. *Dynabook* je prvo spominjanje ideje o ličnim i prenosnim računarima a povodom *Smalltalka* je prvi put lansirana ideja da programski jezik treba da se sastoji ne samo od prevodioca ili interpretera nego i od svih pomoćnih oruđa koja programeru olakšavaju rad.

*Smalltalk* nije jedini objektni jezik. Ubrzo su se pojavili i drugi takvi jezici i samim tim postavilo se pitanje opšte definicije objektnog programiranja. Sledećih sedam uslova razdvajaju objektno orijentisane jezike od drugih tipova jezika.

## MODULARNA STRUKTURA ZASNOVANA NA PODACIMA

Skoro svi jezici ispunjavaju ovaj uslov. Modularnost se uvek može izvesti pomoću potprograma čak i u bejziku ili fortranu. Poželjno bi bilo da su moduli istovremeno i sintaksne i semantičke jedinice jezika. Procedure standardnog paskala su sintaksne ali ne i semantičke jedinice. U *Turbo Pascal* u počev od verzije 4.0 postoje tzv. jedinice (*units*) koje su praktično moduli. Jedinice su nezavisne fizičke i semantičke jedinice na disku i zahvaljujući njima *Turbo Pascal* je mnogo bolji od standardnog paskala. Moduli u *Moduli-2* i paketi (*packages*) u *Adi* su upravo ono što se u teoriji programiranja i podrazumeva pod pojmom modula. Osim po odsustvu neprozirnih tipova (*opaque types*) jedinice *Turbo Pascal* a su praktično iste kao i moduli u *Moduli-2*.

Ovaj uslov, dakle, ne eliminiše skoro nijedan popularan jezik.

## APSTRAKCIJA PODATAKA

Objekti treba da budu konkretizovani u programu kao apstraktni tipovi podataka. Pod apstraktnim tipom podataka podrazumeva se istovremeno zadavanje i opisa podataka i operacija nad njima, pri čemu se do podataka može doći jedino primenom tih operacija. Time se postiže logičko i fizičko razdvajanje podataka i onemogućava se da programer ispravkom jednog dela programa nanese štetu negde drugde. Veći broj programskih jezika udovoljava ovom uslovu a *Ada* i *Module 2* su najpoznatiji primeri. Izgleda čudno, ali se i u fortranu (zahvaljujući naredbi ENTRY) može koristiti apstrakcija podataka. Međutim, standardni *paskal*, *kobol* i *bejzik* uopšte ne ispunjavaju ovaj uslov. Ovde treba precizno reći da se radi o tome da jezik ohrabruje — čak i zahteva — takav pristup, jer se programer u svakom jeziku može uvek zareći da sam poštuje svoje procedure i koristi ih strogo namenski za pristupanje podacima. U praksi se takve nastojanja programera izjalove. Teško je u ime viših principa birati teži put za pisanje i stvaranje programa ako ga sam programski jezik ne podržava i zahteva.

Treći kriterijum je manje konceptualne a više praktične prirode: kako će se objekt stvarati i uništavati, jer računarski resursi uvek su ograničeni.

## AUTOMATSKO UPRAVLJANJE MEMORIJOM

Nekorišćene objekte treba sam programski jezik da briše iz memorije (*garbage collection*), bez eksplicitne programerove intervencije. Na ovom testu većina tradicionalnih programskih jezika pada. *paskal* i *Module 2* obično ne sadrže nikakve sistemske mogućnosti da se memorija oslobađa „starih" objekata, a neki prevodioci za *Adu* to nude kao opciju. S druge strane, svi LISP sistemi to imaju kao standardnu mogućnost, pa je zato dobar deo objektnih jezika svojevremeno pisan upravo kao proširenje LISP-a. Primetimo da se ovde ne radi o sposobnosti jezika da zauzima i oslobađa memoriju (paskal to na primer može naredbama NEW i DISPOSE), već da li neka realizacija jezika to podržava nezavisno i od programera i od programa. Iako do sada nisu napravljeni, nema nikakvih principijelnih razloga protiv postojanja tradicionalnih jezika sa automatskim brisanjem nekorišćenih promenljivih.

Ova tri uslova klasični jezici još koliko toliko i ispunjavaju (ili bi mogli ispunjavati). Sledeći uslov je ono što kardinalno razlikuje objektno orijentisane jezike od svih ostalih.

## MODUL JE TIP I TIP JE MODUL

Tu se ne misli na tipove definisane na nivou jezika poput REAL, CARDINAL, INTEGER, CHAR, STRING i ostalih, već se radi o tipovima podataka koje sam programer definiše. Sintaksna jedinica koja je istovremeno i modul i tip zove se *klasa*. Klasa je opis neke grupe objekata, a objekat je konkretizacija klase. Objektno orijentisani pristup programiranju je uvek hijerarhijski: jedna klasa može sadržati drugu klasu, baš kao što i skup u matematici može sadržavati jedan ili više podskupova. Primeri klasa i podklasa su svuda oko nas: klasa programskih jezika i podklasa objektnih jezika, klasa zgrada i podklasa poslovnih zgrada, zgrada za stanovanje, hotela, sportskih hala i tako dalje.

Znanje počinje klasifikacijom a objektno programiranje počinje svrstavanjem objekata u klase i potklase, odnosno tipove i podtipove podataka. Pri tome je prirodno da za elemente potklase važi sve što i za elemente klase — to ne samo da se mogu primenjivati iste operacije nego da i sve varijable klasa ostanu dostupne u potklasi. Otuda i peti uslov:

## NASLEĐIVANJE KLASA

Klasa može biti definisana kao proširenje (restrikcija) neke druge klase. U upotrebi su termini *roditeljska klasa* i *klasa naslednik* ili neki slični koji odslikavaju hijerarhijsku strukturu i procedura i podataka. Iako se to tako obično ne zove, sa nasleđivanjem se susrećemo bukvalno u svim jezicima. Tako je pojam znaka (CHAR) po pravilu podklasa tipa STRING, a tip INTEGER je podklasa tipa REAL i sl. *paskal* dozvoljava mogućnost definisanje podtipova za „klasu" INTEGER. Time je programiranje veoma olakšano zato što programer primenjuje iste operacije (čak i sintaksu) na različite vrste podataka. Operacija sabiranja se i za tip INTEGER i za tip REAL označava se + i sasvim je prirodno da je to tako. Tu odmah dolazimo i do

fundamentalnog ograničenja klasičnih jezika. Pretpostavimo da smo uveli novi tip podataka po imenu VERYLONGINTEGER. Kao što i ime govori, on omogućava operacije nad celim brojevima dugačkim 20, 30, 100 ili 500 cifara. U klasičnim jezicima na njega se ne može primeniti operacija +, već bi se sabiranje moralo naznačavati kroz pozive procedura ili funkcija. To je već ogromna prepreka efikasnom programiranju. Zato je sledeći uslov zaloga programerske produktivnosti:

## POLIMORFIZAM I KASNO POVEZIVANJE

Naredba se mogu odnositi na objekte iz više klasa, a ista operacija može se realizovati različito od klase do klase. Polimorfizam je osobina objekta da sam razlikuje poruke koje su mu upućene i da na njih reaguje (to jest, da ih izvršava) ako može. Kasno (dinamičko) povezivanje znači da se tokom izvršenja programa ispituje da li u datoj klasi postoji tražena naredba. Možemo zamisliti da u svakom objektu postoji CASE naredba koja aktivno određuje šta sve objekat zna ili ne zna da uradi.

U praksi je sve ovo jednostavno. Pretpostavimo da smo napisali klasu MATRICA, dakle, zadali njenu internu strukturu i operacije koje nad primercima tog tipa možemo vršiti. Matrice su najopštiji brojevi pa je prirodno da se u klasi MATRICA nađu sabiranje, oduzimanje, množenje i ostale aritmetičke operacije. U paskalu ili Moduli-2 bi se pisalo, na primer, sledeće:

```
VAR
    A, B, C : MATRICA
    D, E, F : INTEGER;

A := MATSUB(B,C)
D := E + F;
A := MATSABSKALAR(A,D)
```

Sabiranje je tu zapisano na tri različita načina. Jedino je sabiranje celih brojeva rešeno na uobičajen način, dok dve vrste sabiranja matrica zahtevaju zasebne procedure. To nije ni ekonomično ni prirodno. Kada biste nekom čoveku zadali da izvrši gornja tri sabiranja, on bi u najboljem objektnom maniru prvo pogledao koje operacije treba uraditi u zavisnosti od tipa podataka (dinamičko povezivanje), a zatim bi to i uradio u skladu sa prirodom objekata. U nekom objektnom jeziku umesto izmišljanja posebne sintakse u sva tri sabiranja bi se koristio isti operator, +. Na primer, u Smalltalku:

```
A := MATRICA NEW
B := MATRICA NEW
C := MATRICA NEW
D := INTEGER NEW
E := INTEGER NEW
F := INTEGER NEW

A := B + C
D := E + F
A := A + D
```

To je primer polimorfizma, jer različiti objekti (celi brojevi i matrice u ovom primeru) različito reaguju na istu naredbu +. Sa druge strane, to je i primer za takozvano „preopterećenje operatora" (*operator overloading*). Naredba + je „preopterećena" u smislu da nosi različita značenja u odnosu na različite objekte. Polimorfizam i kasno povezivanje strahovito pojednostavljuju programe: sabiranja svih objekata (za koje to ima smisla) slivaju se u naredbu +, a slično je i sa svim oduzimanjima, množenjima, deljenjima (ako ona ima smisla), crtanjima i tako dalje.

Ovih šest uslova treba da ispunjava svaki objektni jezik. Tu se može dodati i poslednji, neobavezni, ali i veoma koristan uslov:

## VIŠESTRUKO I PONOVLJENO NASLEĐIVANJE

To bi značilo da je jedna klasa istovremeno podklasa za dve ili više klasa, odnosno da jedna klasa može biti dva ili više puta naslednik iste klase. Lepo bi bilo definisati prozor na ekranu da bude istovremeno i tipa PROZOR i tipa MULTITASKING, pa da se njemu obraćamo kao prozoru u kojem se vidi neki proces istovremeno sa drugim procesima u programu. Jezici koji od samog početka nisu predvideli višestruko nasleđivanje kasnije ga teško mogu stvoriti na sintaksnom nivou. Tako se došlo do kuriozne situacije da Smalltalk, koji je bio prvi pravi objektni jezik, ne ispunjava ovaj kriterijum, dok ga C++ ispunjava. (Što ne znači da je C++ bolji od Smalltalka. Ahilova peta jezika C++ je kriterijum 3 – raspolaganje memorijom nije automatsko već je „skupljanje đubreta" ostavljeno na volju programeru.) Jedini objektni jezik koji bi ispunjavao sve navedene uslove je *Eiffel*, ali isključivo zato što je i napisan sa ciljem da ih ispunjava.

Bilo bi pogrešno shvatiti da se korišćenjem objektnih naredbi u nekom jeziku automatski dobija program koji je i sam objektno orijentisan! Da bi neki program bio takav, mora da ispunjava bar prvih šest uslova, što znači da u tom programu postoje klase, metodi, nasleđivanje i drugo. U programu za crtanje to bi značilo da korisnik može da napravi klasu objekata koje će da crta, da iz postojećih klasa npr. opšteg modela kuće nasleđivanjem može da zadaje nove grafičke objekte (npr. da zada broj zidova, visinu kuće itd). U programu za rad sa bazama podataka, to bi značilo da korisnik raspolaže opštim modelom npr. maske za unos podataka o materijalnom poslovanju i da onda može da napravi („nasledi") iz te maske neku drugu masku, npr. za unos podataka o materijalnom poslovanju u nekoj konkretnoj radnoj jedinici.

Popularni programski jezici su vremenom počeli da ispunjavaju neke od gornjih sedam uslova. To se dobro vidi iz istorije samog *Turbo Pascal*-a. Od verzije 4.0 u njemu je ispunjen prvi uslov, modularnost, jer jedinice mogu biti pravi moduli. Samo od programera zavisi da li će jedinica postati modul, jer i dalje je moguće pisati velike monolitne programe u *Turbo Pascal*-u. Jedinica se deli na dva dela, naredbama **interface** i **implementation**, što omogućava apstrakciju podataka. Tako verzija 4.0, uz uobičajene procedure i funkcije iz standardnog *paskala*, zadovoljava prva dva uslova. Treći uslov, automatsko brisanje nepotrebnih podataka iz memorije, u *Turbo Pascal*-u 5.5 nije ispunjen. Programer se za to mora pobrinuti sam, parovima naredbi NEW i DISPOSE, odnosno naredbama MARK i RELEASE. Četvrti, peti i šesti uslov su ispunjeni u verziji 5.5, ali sa nekim ograničenjima u odnosu na čiste objektno orijentisane jezike. Sedmi uslov, višestruko nasleđivanje, u *Turbo Pascal*-u 5.5 nije zastupljen, ali se može simulirati. Zato je *Turbo Pascal* 5.5 hibridni objektni jezik: na njemu se mogu pisati programi koji koriste objektno orijentisane naredbe i koncepte, ali se mogu pisati i programi na uobičajen, „paskalski" način. Time se čuva vrednost već napisanih programa, a programerima se pruža šansa za upoznavanje i praktičnu primenu objektnog programiranja.

Evo kako *Turbo Pascal* 5.5 ispunjava gornje uslove. Pretpostavićemo poznavanje elemenata standardnog paskala.

# SINTAKSA PROGRAMA

U terminologiji *Turbo Pascal*-a modul je zajedničko ime za programe i jedinice.

Program je sličan deklaraciji procedure, osim što postoje zaglavlje i opciona naredba **uses**. Zaglavlje programa se sastoji iz službene reči **program**, imena programa i opcionih parametara. Zaglavlje programa, ako uopšte postoji, služi samo u dekorativne svrhe i *Turbo Pascal* ga ignoriše. U standardnom paskalu program je jedna (velika) procedura, a pošto procedure imaju parametre, logično sledi da i naredba **program** treba da ih ima. Parametri naredbe **program** su imena datoteka kojima program treba da ima pristup dok radi. Kao minimum definišu se dve standardne datoteke **Input** i **Output** za standardni ulaz i izlaz. U DOS-u se unapred zna koje su to datoteke, tj. ulazno-izlazni uređaji, tako da nema nikakve potrebe da se uvek i stvarno navode. Osim toga, funkcijom **ParamStr** mogu se učitati parametri sa komandne linije DOS-a, što i ostale parametre u eventualnoj naredbi **program** čini besmislenim. Zato se i sama naredba **program** uvek može izostaviti.

Naredba **uses** zadaje sve jedinice koje program koristi, kako jedinice koje se koriste direktno, tako i jedinice koje koriste druge jedinice. Jedinica **system** sadrži naredbe i tipove paskala i uvek se koristi bez posebnog navođenja u naredbi **uses**. Osim jedinice **system**, *Turbo Pascal* se isporučuje i sa drugim standardnim jedinicama poput **Printer**, **DOS** i **Crt**. One se ne koriste automatski, već se moraju poimence uključivati naredbom **uses**.

Najjednostavniji glavni program glasi ovako:

```
program tp1;
begin
end.
```

# SINTAKSA JEDINICE

Jedinice su osnova modularnog programiranja u *Turbo Pascal*-u. Kroz njih se stvaraju biblioteke procedura i funkcija. Te biblioteke možemo dodavati različitim programima, a da tekst programa nije nužno prisutan. Tako velike programe delimo na manje delove, tj. na module, a njih je znatno lakše pisati, testirati i menjati.

Prevodilac prvo pretražuje jedinice koje su već učitane u memoriju (te jedinice se učitavaju zajedno sa *Turbo Pascal*-om iz datoteke TURBO.TPL). Ako tu nema navedene jedinice, prevodilac pretpostavlja da je na disku. U tom smislu čini i dodatnu pretpostavku da je ime jedinice istovetno sa imenom datoteke u kojoj je ta jedinica sa prezimenom TPL. Ako je nema ni u tekućem imeniku, traži se u drugim imenicima.

Standardne jedinice uzete zajedno zapravo definišu *Turbo Pascal* kao programski jezik.

U verziji 5.5 u jedinicama se mogu definisati i objekti. Ne postoji nikakva sintaksna veza između objekata i modula jer:

- tekst nekog objekta može sav biti u jednoj jedinici,
- više objekata može biti u jednoj jedinici, ali i
- tekst jednog objekta može biti raspoređen na nekoliko jedinica na disku.

Preporučljivo je da svi metodi jedne klase fizički budu u jednoj jedinici.

Jedinica počinje zaglavljem koje se sastoji iz službene reči **unit** i imena jedinice. Ime jedinice se koristi u naredbi **uses**. Ne smeju postojati dva ista imena jedinica u istom programu. Ime jedinice i ime datoteke u kojoj se ona nalazi moraju se poklapati, jer kasnije *Turbo Pascal* po tome imenu razaznaje datoteke na disku.

Jedinica se sastoji iz tri dela i svaki počinje određenom rezervisanom rečju: **interface**, **implementation** i **begin**. Ova poslednja reč može se izostaviti.

Deo jedinice koji počinje sa **interface** deklariše konstante, tipove, varijable, procedure i funkcije javnog karaktera. Reč „javni" ovde znači da se svi elementi definisani kao **interface** mogu koristiti i u glavnom programu i u nekim drugim jedinicama. **Interface** bi se moglo prevesti kao i „izvozni" deo, jer se kroz njega podaci, procedure i funkcije „izvoze" u ostatak programa, drugim modulima.

U **interface** delu jedinice navode se deklaracije procedura i funkcija. U sintaksnom pogledu te deklaracije ponašaju se isto kao i da je upotrebljena službena reč **forward** – osim što je izostavljena. Telo takve procedure ili funkcije obavezno se mora navesti u **implementation** delu jedinice, sa izuzetkom deklaracija tipa **inline**, kod kojih se telo procedure ili funkcije – po prirodi stvari – navodi odmah.

Deklaracije se van jedinice upotrebljavaju kao da su iz glavnog programa.

Reč **implementation** označava izvršni deo jedinice, jer se u njemu nalaze tela procedura i funkcija naznačenih u **interface** delu jedinice. Izvršni deo takođe može deklarisati konstante, tipove, varijable, procedure i funkcije. Sve što je definisano u izvršnom delu ostaje privatno (lokalno) za datu jedinicu, u smislu da za ostatak programa van jedinice ne postoji.

Smisao podele jedinice na javne i privatne podatke i potprograme je tzv. lokalizacija podataka (*information hiding*). Idealno bi bilo da u jednoj jedinici postoje samo operacije nad jednom strukturom podataka. To znači da je sve što se te strukture podataka tiče smešteno u jednu datoteku. Pošto su izmene strukture podataka mnogo ređe od izmena naredbi koje nad njima dejstvuju, ostatak programa ništa ne zna o strukturi podataka, već samo o operacijama nad njom. Čitavi delovi programa mogu se menjati bez izmena po ostatak programa. Takvom organizacijom se izbegava „domino efekat", tj. da izmena jednog modula „ruši" i sve ostale module.

Deo za postavljanje početnih vrednosti jedinice je – fizički gledano – poslednji deo jedinice. Sastoji se ili od rezervisane reči **end** (tada jedinica i nema posebne naredbe za postavljanje) ili od bloka naredbi **begin**...**end** koje se izvršavaju redosledom pisanja i time postavljaju početne vrednosti jedinice.

Javni potprogrami iz izvršnog dela aktiviraju se navođenjem njihovog imena. Konačno, lokalne potprograme aktiviraju javni potprogrami po potrebi.

Naredba **uses** u modulu treba samo da navede jedinice koje taj modul direktno koristi. Navođenje imena jedinice u naredbi **uses** automatski izvršava pripremni deo te jedinice. To se obično koristi za početno postavljanje konstanti koje se u jedinici koriste. Bez toga bi bilo vrlo teško isprogramirati generisanje slučajnih brojeva.

Prilikom prevođenja *Turbo Pascal* mora biti u mogućnosti da pronađe sve jedinice od kojih modul zavisi, bilo direktno, bilo indirektno. Promena deklaracija u **interface** delu jedinice povlači za sobom ponovno prevođenje svih modula koji koriste tu jedinicu. Izmene u druga dva dela jedinice nemaju takav efekat.

**Interface** delu jedinice se tokom prevođenja pridodeljuje verzija, tj. kontrolni broj. Ako se kontrolni brojevi dvaju jedinica ne poklope kod sledećeg prevođenja, znači da postoje izmene pa se ponovnim prevođenjem verzije jedinica usklađuju.

Ako se naredba **uses** nađe u izvršnom delu jedinice, onda se unutrašnji detalji još više skrivaju. Međutim, još važnija je činjenica da se na taj način omogućava stvaranje uzajamno nezavisnih jedinica. **Interface** delovi jedinica prevode se nezavisno od **implementation** delova.

Najjednostavnija jedinica glasi ovako:

```
unit j1
interface
implementation
end
```

Dok se rezervisana reč **program** može izostaviti na početku glavnog programa, za jedinice to ne važi: rezervisana reč **unit** mora biti prvi simbol u jedinici.

Jedinica **j1** ima dva od tri moguća dela; sledeća jedinica, **j2**, ima i pripremni deo, tj. jedinici **j1** dodata je još samo reč **begin**.

```
unit j2
interface
implementation
begin
end
```

Ilustrujmo aktiviranje pripremnog dela jedinice. Program **tp1** sada sadrži i naredbu **uses** kojom se priključuje jedinica **j3**. Iako nema nijednu izvršnu naredbu, ipak **tp1** prouzrokuje izvršenje **writeln** naredbe iz pripremnog dela jedinice **j3**.

```
program tp1;
uses j3;
begin
end.

unit j3;
interface
implementation
begin
    writeln('Ovo je pripremni deo');
end
```

Sledeći primer prikazuje potpunu lokalnost podataka koju omogućava **implementation** deo jedinice.

```
unit j4;
interface
implementation
var horStrana : integer
procedure pHorStrana(novo : integer);
begin
    horStrana := novo;
end;
end
```

Varijabla **horStrana** je normalno deklarisana u **var** naredbi i sa sintaksne strane nema nikakvih problema. Procedura **pHorStrana** (postavljanje horizontalne strane) je takođe legalna i deluje nad promenljivom **horStrana**. Jedini problem je što su i **horStrana** i **pHorStrana** deklarisani u **implementation** delu jedinice **j4** i kao takvi – nepoznati su izvan tog **implementation** dela. U jedinici **j4** kako je gore data, i promenljiva i procedura su sasvim beskorisne: nijedna druga procedura ili funkcija iz **j4** ne zove proceduru **pHorStrana** pa je promena vrednosti u **horStrana** nemoguća. Takođe nije moguće zvati proceduru **pHorStrana** iz glavnog programa, jer **pHorStrana** nije izvezena kroz **interface** deo. Za to bi trebalo deklaraciju procedure prebaciti u **interface** deo, na primer, ovako:

```
unit j5
interface
procedure pHorStrana(novo : integer);
function dHorStrana : integer;
implementation
var horStrana : integer;
procedure pHorStrana(novo : integer);
begin
    horStrana := novo;
end
function dHorStrana;
begin
    dHorStrana := horStrana;
end;
end
```

Iskoristili smo priliku da dodamo i funkciju **dHorStrana** koja vraća vrednost privatne promenljive **horStrana**. I procedura **pHorStrana** i funkcija **dHorStrana** sada su u **interface** delu, pa ih ostali moduli mogu koristiti. Deklaracije procedura i funkcija u **interface** delu ponašaju se kao u **forward** naredbi, osim što se ova službena reč ne piše. Kod **forward** deklaracija možemo oba puta pisati celu listu parametara ali i ne moramo. Primera radi, u jedinici **j5** cela deklaracija procedure **pHorStrana** je ponovljena u **implementation** delu, a za funkciju **dHorStrana** deklaracija nije ponovljena. U praksi najzgodnije je ne ponavljati deklaracije jer ako se u **interface** delu nešto promeni, to ćemo morati ručno da ponovimo i u odgovarajućoj deklaraciji u **implementation** delu.

Evo i glavnog programa koji – ovoga puta bez problema – poziva proceduru i funkciju iz **j5**:

```
program tp1
uses j5;
begin
    pHorStrana(3);
    writeln('Vrednost horizontalne strane
je ', dHorStrana);
end
```

Glavni program postavlja vrednost na 3 i u naredbi za ispis je očitava. Nigde nije rečeno da se vrednost promenljive **horStrana** postavlja na 3 niti da se upravo vrednost te promenljive očitava funkcijom **dHorStrana**. Ovaj glavni program nema načina da dopre do promenljive **horStrana**, jer je ona definisana u **implementation** delu jedinice **j5**. Što se tiče glavnog programa, radi se o nekoj apstraktnoj vrednosti koja se postavlja pozivom jedne procedure ili funkcije i čija se vrednost očitava nekom drugom procedurom (ili funkcijom). Zato kažemo da jedinica **j5** definiše apstraktan tip. To je dragoceno svojstvo za svaki program. Zaista, unutrašnjost jedinice **j5** možemo dalje menjati a da glavni program (ili bilo koji drugi korisnik apstraktnog tipa iz jedinice **j5**) nastavi da i dalje radi bez izmena.

## APSTRAKCIJA PODATAKA

Deklaracije u **implementation** delu jedinice omogućavaju *Turbo Pascal*-u da ispuni drugi uslov – apstrakciju podataka. Od programera zavisi da li će – i koliko – koristiti blagodeti apstraktnih tipova. Pri tome programeru ostaje otvoren i drugi put – uvođenje deklaracija u **interface** deo jedinice. U nekim slučajevima to se mora raditi ali se time ruši osnovna ideja apstraktnih tipova podataka, tj. da se do njih može doći isključivo preko procedura i funkcija. Razmotrimo jedinicu **j6**:

```
unit j6
interface
var horStrana : integer
procedure pHorStrana(novo : integer);
function dHorStrana : integer
implementation
procedure pHorStrana(novo : integer
begin
    horStrana := novo
end
function dHorStrana
begin
    dHorStrana := horStrana
end
end
```

Sve je isto osim što je linija sa deklaracijom premeštena u

interface deo. Time je promenljiva **horStrana** postala dostupna modulima izvan jedinice **j6**. Sledeći program to ilustruje:

```
program tp1;
uses j6;
begin
     pHorStrana(3);
     writeln('Vrednost horizontalne strane je ', dHorStrana);
     horStrana := 6;
     writeln('Direktno izmenjena vrednost je ', j6.dHorStrana);
     j6.horStrana := 8
end.
```

Prve dve linije su iste, jer postojeće procedure i funkcija i dalje važe. Treća linija je novost: promenljivu **horStrana** direktno postavljamo na 6 iako nije deklarisana u glavnom programu. U poslednje dve linije ilustruje se način dolaženja do imena u jedinici – ispred možemo postaviti ime jedinice i tačku. Tako je **j6.horStrana** isto što i **horStrana**, a **j6.dHorStrana** je isto što i **dHorStrana**.

Sa stanovišta apstrakcije podataka ova sintaksna mogućnost ne valja. Programer može neposredno da pristupi svakom podatku iz **interface** dela i da ga promeni ili čita nezavisno od protokola kojem su bili podaci podvrgnuti, ako se koriste druge procedure i funkcije. Kao što ćemo videti, objekti se uvek moraju definisati u **interface** delu jedinice, tako da *Turbo Pascal* jeste objektno orijentisan jezik – koji apstrakciju podataka ne podržava automatski.

Naravno, definisati apstraktni tip umesto jedne jedine promenljive je preterivanje, u paskal su već ugrađene naredbe za direktan rad sa promenljivama. U praksi se obično definiše neki slog (**record**) i nad njim izvode odgovarajuće operacije. Primera radi, opišimo pravougaonik kao slog:

```
unit j7;
interface
type
   Pravougaonik = record
     horStrana,
     verStrana : integer;
     end; { Pravougaonik }
  procedure pHorStrana(novo : integer);
  function dHorStrana : integer;
  procedure pVerStrana(novo : integer);
  function dVerStrana : integer;
  procedure wr;
  procedure dPravougaonik(var pr : Pravougaonik);
implementation
var p : Pravougaonik;
procedure pHorStrana(novo : integer);
begin
     with p do horStrana := novo;
end;
function dHorStrana;
begin
     dHorStrana := p.horStrana;
end;
procedure pVerStrana;
begin
     p.verStrana := novo
end;
function dVerStrana;
begin
     dVerStrana := p.verStrana
end;
procedure wr;
begin
     writeln('Horizontalna strana pravougaonika je ', p.horStrana);
     writeln('Vertikalna strana pravougaonika je ', p.verStrana);
end; { wr }
procedure dPravougaonik;
begin
     pr := p;
end; { dPravougaonik }
end.
```

Osnovna struktura je sada slog **Pravougaonik**, koji se sastoji iz dve promenljive: **horStrana**, **verStrana**. Broj osnovnih procedura i funkcija se udvostručio: procedura **wr** prikazuje stanje pravougaonika na ekranu dok procedura **dPravougaonik** vraća vrednost tipa **Pravougaonik** kroz promenljivu **pr**. Sve te procedure deluju nad primerkom tipa **Pravougaonik**, privatnom promenljivom **p** (privatna je jer je deklarisana tek u **implementation** delu jedinice). Glavni program bi mogao izgledati ovako:

```
program tp1;
uses j7, Crt;
var p : Pravougaonik;
begin
     ClrScr;
     pHorStrana(3);
     pVerStrana(4);
     wr;
     dPravougaonik(p);
     p.horStrana := 11;
     wr;
     writeln('p.horStrana =', p.horStrana);
end.
```

U naredbi **uses** su dve jedinice: jedna koju smo sami napisali (**j7**) i druga, **Crt**, koja je standardna jedinica *Turbo Pascal*-a. Iz ove druge koristimo proceduru **ClrScr** da bi jasno razdvojili izlazne rezultate programa od ostatka ekrana. Novina u ovom programu je deklarisanje promenljive p tipa **Pravougaonik** – taj tip je uvezen iz jedinice j7. Poziv procedure **dPravougaonik** postavlja **p** na tekuću vrednost lokalne promenljive **p** iz jedinice. Kada potom direktno postavimo horizontalnu stranu pravougaonika na 11, vrednost privatne promenljive **p** se ne menja, o čemu svedoči drugo izvršenje procedure **wr**.

## VRAĆANJE TIPOVA

Procedura **dPravougaonik** zaslužuje poseban komentar. Njome se vraća vrednost promenljive p, pa bi bilo logično očekivati funkciju umesto procedure, odnosno trebalo bi očekivati ovakvo zaglavlje u jedinici j7:

```
function dPravougaonik : Pravougaonik;
```

s tim da u programu pišemo

```
p := dPravougaonik;
```

ili nešto slično. Međutim, to u paskalu ne može, jer funkcija može vratiti najviše četvorobajtnu vrednost a nikako sam tip (u ovom slučaju **Pravougaonik**). Rešenje je da funkcija vraća pokazivač na tip. U tom slučaju bi jedinica **j8** mogla izgledati ovako:

```
unit j8;
interface
type
   PravougaonikPok = ^Pravougaonik;
   Pravougaonik = record
     horStrana,
     verStrana : integer
     end; { Pravougaonik }
```

```
      dSo
    o pV
    dVe
      ai
    r
   dure
   procedure brisP;
   procedure pSvaStr
nikPok);
   function dSvaStrana : PravougaonikPok;
implementation
var p : Pravougaonik;
    pPok : PravougaonikPok;
procedure pHorStrana(novo : integer);
begin
    with pPok^ do horStrana := novo
end;
function dHorStrana
begin
    dHorStrana := pPok^.horStrana;
end;
procedure pVerStrana;
begin
    pPok^.verStrana := novo
end
function dVerStrana;
begin
    dVerStrana := pPok^.verStrana
end
procedure wr;
begin
    writeln('Horizontalna strana pravougao
nika je ', pPok^.horStrana);
    writeln('Vertikalna strana pravougaoni
ka je ', pPok^.verStrana);
end; { wr }
function vratiP;
begin
    vratiP := pPok
end { vratiP }
procedure noviP;
begin
    new(pPok)
end;
procedure brisaP
begin
    dispose(pPok);
end
```

```
   re pSvaStrana;

    pok := stranaPok;
nd { pSvaStrana }
unction dSvaStrana
begin
    dSvaStrana := pPok;
end; { dSvaStrana }
end
```

Novi tip je **PravougaonikPok**, tj. pokazivač na **Pravougaonik**. Prve četiri procedure i funkcije su iste, ali sada postoji funkcija **vratiP** (umesto predašnje procedure **dPravougaonik**) koja vraća traženi pokazivač. I sve ostale procedure i funkcije su napisane tako da koriste pokazivač **pPok**. U tekstu i dalje figuriše promenljiva p ali se više ne koristi.

Glavni program u svojoj najjednostavnijoj verziji može izgledati ovako:

```
program tpl
uses j0, Crt;
var p : PravougaonikPok;
begin
    ClrScr;
    new(p);
    pHorStrana(3);
    pVerStrana(4)
    wr
    p := vratiP;
    p^.horStrana := 12;
    dispose(p)
end
```

Pokazivač p je dinamička promenljiva, što znači da se za nju odvaja memorijski prostor ne tokom prevođenja nego tek tokom izvršenja programa. Zato moramo naredbama **new** i **dispose** eksplicitno zauzeti, odnosno, osloboditi memoriju. To nije dovoljno da definiše značenje pokazivača, tek naredbom p := **vratiP** usmeravamo **p** na tip **Pravougaonik**. Elementima tog tipa obraćamo se imenom pokazivača iza kojeg dolazi strelica u vis: p^.

U *Turbo Pascal*-u 5.5 od automatskog upravljanja memorijom nema ništa: sve moramo eksplicitno narediti sa **new** i **dispose**. Za svaku izvršenu naredbu **new** mora se izvršiti odgovarajuća naredba **dispose**. To može predstavljati praktičan problem, jer ako npr. **dispose** piše u tekstu programa nije uvek sigurno da će se i izvršiti. Najteže je testirati i verifikovati programe koji sadrže pokazivače.

# POJAM OBJEKTA

Za sve dosadašnje primere karakteristično je da na sintaksnom nivou nema nikakve veze između naredbi koje upućujemo nekom apstraktnom tipu podataka i promenljivih koje pripadaju tom tipu. Kada u programu napišemo **pHorStrana(3)** ili **pVerStrana(4)** ne vidi se na koju promenljivu se to odnosi. U jedinici **j0** jeste definisan tip **Pravougaonik** ali procedure **pHorStrana** i **pVerStrana** mogu delovati i nad bilo kojim drugim tipom. Samo programerovi misaoni procesi povezuju tip **Pravougaonik** sa **pHorStrana** i **pVerStrana** – iz teksta programa se ta veza ne vidi. Jedinica (modul) u *Turbo Pascal*-u može predstavljati tip podataka ali taj tip nije modul. Da bi modul bio tip i da bi tip bio modul potrebno je nadgraditi pojam sloga (**record**) i uvesti pojam objekta (**object**).

Objekat je slog u kojem se mogu naći procedure i funkcije koje operišu nad elementima sloga. Razlike u sintaksi su minimalne. Slog se piše kao

```
TYPE Pravougaonik = record
        horStrana,
        verStrana : integer
     end
```

```
TYPE Pravougaonik = object
        horStrana,
        verStrana : integer;
     end
```

Takav objekat je novi tip podataka i primerke tog tipa definišemo naredbom VAR:

```
VAR pr : Pravougaonik
```

Poljima takvog sloga može se prići na uobičajen način, naredbom WITH ili korišćenjem tačke:

```
pr.horStrana := 10
with pr do begin
  horStrana := 25;
  verStrana := 456;
  end
```

U jedinici j9 tip Pravougaonik napisan je kao objekat

```
unit j9;
interface
type
  Pravougaonik = object
      horStrana,
      verStrana : integer;
      procedure pHorStrana(novo : integer);
      function dHorStrana : integer;
      procedure pVerStrana(novo : integer);
      function dVerStrana : integer;
      procedure wr;
      procedure dPravougaonik(var pr : Pravou
gaonik);
      end  { Pravougaonik }
implementation
procedure Pravougaonik.pHorStrana(novo : in-
teger);
begin
      horStrana := novo;
end;
function Pravougaonik.dHorStrana;
begin
      dHorStrana := horStrana;
end;
procedure Pravougaonik.pVerStrana;
begin
      verStrana := novo
end;
function Pravougaonik.dVerStrana;
begin
      dVerStrana := verStrana;
end;
procedure Pravougaonik.wr;
begin
      writeln('Horizontalna strana pravougao
nika je ', horStrana);
      writeln('Vertikalna strana pravougaoni
ka je ', verStrana);
end; { wr }
procedure Pravougaonik.dPravougaonik;
begin
      pr := self
end; { dPravougaonik }
end
```

Sastavni delovi klase **Pravougaonik** su promenljive **horStrana** i **verStrana**, ali ne samo one već i sve procedure i funkcije koje definišu pravougaonik kao apstraktan tip.

Iako se upotrebljava službena reč **object**, u jedinici **j9** zapravo smo definisali klasu **Pravougaonik**, a kada se službenom rečju **var** definiše primerak te klase, možemo govoriti o objektu kao o konkretizaciji klase. Procedure i funkcije iz jedne klase nazivamo metodima te klase.

Kao što se u jedinici može zadati nekoliko slogova, tako se u stoj jedinici može napisati nekoliko objekata. Da bi se znalo kojoj klasi pripadaju metodi, ispred imena metoda u zaglavlju obavezno se ubacuje ime klase i razdvaja tačkom od imena metoda. Umesto function dHorStrana mora se pisati **function Pravougaonik.dHorStrana**. Ako je ime klase dugačko (kao u ovom slučaju) možemo prvo izostaviti ime klase a zatim ga označiti kao blok i prečicom **Ctrl-K C** ubacivati u zaglavlje metoda.

U prethodnim primerima bilo je obavezno definisati lokalnu promenljivu npr. **p** nad kojom su delovale procedure i funkcije apstraktnih tipova. U definiciji klase toga nema – lokalne promenljive tipa **Pravougaonik** se podrazumeva, baš kao da su svi metodi u nekoj (pomnoj) **with** naredbi. Ako ipak želimo (ili moramo) upotrebiti tu nevidljivu lokalnu promenljivu, onda se koristi nova službena reč **self**. Ona se relativno retko koristi, mada je metod **dPravougaonik** sadrži jer nema drugog načina da se odjednom vrati vrednost cele promenljive. Zaglavlje ovog metoda interesantno je i zato što se u njemu pojavljuje **tip Pravougaonik**, tj. u definiciji klase pojavljuje se varijabla iste te klase. Ovakvo kružno pozivanje na isti tip u *Turbo Pascal* u ne predstavlja nikakav problem.

Idealno bi bilo imati metod **horStrana** koji bi vraćao vrednost promenljive **horStrana**, ali to nije mogućno. Pravila paskala zabranjuju da se ime promenljive koristi kao ime metoda.

U izvesnom smislu metodi i promenljive u klasi su ravnopravni. To se vidi i kroz sintaksu: sa *objekat.promenljiva* dolazi se do vrednosti promenljive u objektu, a sa *objekat.metod* aktivira se metod. Kažemo još da se tako objektu šalje poruka. Eventualni parametri poruke navode se u zagradama iza imena metoda, u skladu sa paskalskim konvencijama. Sledeći program to ilustruje:

```
program tp1;
uses j9, Crt;
type pravougaonikPok = ^Pravougaonik;
     znakPok = ^char;
var p : Pravougaonik;
    pok, pok2 : pravougaonikPok;
    zPok : znakPok;
begin
     ClrScr;
     p.pHorStrana(3);
     p.pVerStrana(4);
     p.wr;
     p.horStrana := 34;
{ Primeri za pokazivace i mogucnosti naredbe
NEW }
     new(pok);
     dispose(pok);
     pok2 := new(pravougaonikPok);
     zPok := new(znakPok);
end
```

Promenljiva **p** je objekat tj. primerak klase **Pravougaonik**. Objektu p upućene su poruke **pHorStrana** sa argumentom 3, **pVerStrana** sa argumentom 4 i **wr**, bez argumenata.

Kada se pogrešna ili nepostojeća poruka pošalje objektu *Turbo Pascal* već tokom prevođenja šalje izveštaj o sintaksnoj grešci. Bez uvođenja pojma klase i objekta takva preciznost bi bila nemoguća. Klase koje programer piše sintaksno su ravnopravne sa klasama koje su ugrađene u sam jezik (**integer**, **real** i druge).

Promenljivima objekta možemo pristupati direktno, što se vidi iz reda iznad komentara.

## PRIPREMA OBJEKTA ZA RAD

Treba uvek postavljati početno stanje objekta, tj. promenljivama objekta treba pridodeliti neku vrednost na početku programa. U gornjem programu to smo uradili sa dve poruke u drugom i trećem redu. Početne vrednosti možemo pridodeliti i samo jednim metodom. Uobičajeno je da se on zove **init**, skraćeno od engleske reči **initialization** (početno postavljanje). U klasi **Pravougaonik** takvog metoda nema ali ga lako možemo dopisati.

Pokazivači na objekte postoje i možemo ih koristiti na uobičajen način. Promenljiva **pok** je pokazivač na objekat tipa **Pravougaonik** i sa **new(pok)** ga pripremamo za rad. Sledeća naredba **dispose(pok)** odmah ukida **pok**. Do verzije 5.5, **new** je uvek bila procedura, ali sada može da bude i funkcija. Poslednja dva reda prikazuju **new** kao funkciju: prvo kao pokazivač na objekat a onda – opštije – kao pokazivač na ugrađeni tip **char**.

Ovim se mogućnosti naredbe **new** ne iscrpljuju: u verziji 5.5 **new** može sadržati i poziv konstruktora, a dispose može sadržati i poziv destruktora. Pre nego uvedemo pojmove konstruktora i destruktora moramo se upoznati sa nasleđivanjem klasa u *Turbo Pascal* u 5.5

## NASLEĐIVANJE KLASA

```
unit j10
interface
type
    pravougaonik = [illegible]
      horStrana, verStrana : integer
      end { pravougaonik }
    pravougaonikObim = record
      pr : pravougaonik
      obim : [illegible]
      end { pravougaonikObim }
implementation
end
```

```
procedure Pravougaonik.pHorStrana
begin
      horStrana := novo
end
procedure Pravougaonik.pVerStrana
begin
      verStrana := novo
end
procedure pravougaonikObim.pObim
begin
      obim := 2 * (horStrana + verStrana)
end { pObim }
end
```

Slog **pravougaonik** koristimo da u sledećem slogu **pravougaonikObim** definišemo tip promenljive pr. Dodali smo i promenljivu **obim** koja će sadržati izračunati obim pravougaonika. Promenljiva **pr** sadrži sve podatke kao i **pravougaonik** pa možemo reći da je slog **pravougaonikObim** *nasledio* slog **pravougaonik**, odnosno da ga je obogatio promenljivom **obim**. Možemo se izraziti i ovako: **pravougaonikObim** je isto što i **pravougaonik**, osim što mu je dodata još jedna promenljiva. Sve su to različiti načini da se izrazi nasleđivanje podataka. Objekti izražavaju nasleđivanje na sintaksnom nivou. Evo gornjeg primera u objektno orijentisanoj notaciji

```
unit j11
interface
type
    [illegible]
end
```

Službena reč **object** može imati i argument – klasu koja se nasleđuje. Klasa **pravougaonikObim** nasleđuje klasu **Pravougaonik** jer je taj argument u definiciji objekta **pravougaonikObim** baš **Pravougaonik**. Nasleđuje se *sve* – u gornjem primeru to su samo promenljive **horStrana** i **verStrana** a da je bilo metoda u klasi **Pravougaonik** i oni bi automatski bili dostupni objektima tipa **pravougaonikObim**. Evo najjednostavnijeg primera nasleđivanja

```
[illegible]
```

Jedinica j12 dopunjuje klase **Pravougaonik** i **pravougaonikObim** metodima za postavljanje vrednosti promenljivih **horStrana** i **verStrana** i metodom **pObim** za izračunavanje obima pravougaonika. Prva dva metoda prepisana su iz ranijih definicija klase **Pravougaonik** dok je metod **pObim** nov. Objektima tipa **pravougaonikObim** možemo slati sve tri poruke (**pHorStrana**, **pVerStrana**, **pObim**) dok objektima tipa **Pravougaonik** možemo slati samo prve dve poruke. Slično, objekat tipa **pravougaonikObim** sadrži tri promenljive (**horStrana**, **verStrana**, **obim**) a objekat tipa **Pravougaonik** sadrži samo prve dve promenljive. Sledeći program sve to ilustruje

```
[illegible]
```

Promenljiva **po** je objekat klase **pravougaonikObim**. Na nju primenjujemo metode nasleđene iz klase **Pravougaonik**, radi postavljanja početnog stanja. Zatim porukom **po.pObim** izračunavamo vrednost promenljive **obim** i prikazujemo je naredbom **writeln**. U sledećem redu vidi se da je promenljiva **horStrana** zaista deo objekta **po** jer je neposredno postavljamo na 34. Poslednje dva reda programa izračunavaju i štampaju **obim** sa tom novom vrednošću.

Nasleđivanje klasa bitno olakšava pisanje programa – pod uslovom da postoje klase od kojih se može nasleđivati. Uz *Turbo Pascal* 5.5 se isporučuje veliki broj klasa i treba biti upoznat sa njima (jedno ovog teksta posvećen je korišćenju tih klasa). Drugi izvor klasa su nezavisni prodavci softvera, a mnogo ideja se može dobiti i iz stručne literature. Konačno, klase možemo pisati i sami. Odakle god dobili klase, bitno je da se sa njima temeljno upoznamo. To zahteva izvesno vreme za učenje ali je na duži rok veoma isplativo: lakše je koristiti već napisano nego stalno pisati iznova. Nasleđivanje nas oslobađa obaveze da „izmišljamo točak" ali podrazumeva da znamo kakvi „točkovi" već postoje.

## REDEFINISANJE METODA

Uvodimo još jednu klasu, **Kvadrat**. Posvetićemo joj posebnu jedinicu **j13**:

```
[illegible]
[illegible]
[illegible]
[illegible]
  Kvadrat = object(pravougaonikObim)
    procedure pHorStrana(novo : integer);
    procedure pVerStrana(aaa, bbb : integer);
    procedure pObim;
    end; { kvadrat }
implementation
procedure Kvadrat.pHorStrana;
begin
end;
procedure Kvadrat.pVerStrana;
begin
   pravougaonikObim.pVerStrana(aaa);
   pravougaonik.pHorStrana(bbb)
end;
procedure Kvadrat.pObim;
begin
   obim := 4 * horStrana
end; { pObim }
end.
```

Klasa **Kvadrat** nasleđuje klasu **pravougaonikObim** iz jedinice **j12**. Time ona automatski nasleđuje i klasu **Pravougaonik**. To znači da klasa **Kvadrat** sadrži tri promenljive: **horStrana**, **verStrana** i **obim**. Umesto da nasledi i sve metode iz „gornjih" klasa, klasa **Kvadrat** *redefiniše* sva tri postojeća metoda. Metod **Kvadrat.pHorStrana** ne sadrži nikakve naredbe. To može izgledati čudno ali ima svog opravdanja: klasi **Kvadrat** i dalje možemo slati poruku **pHorStrana** bez obzira što ona nema efekta.

Zaglavlje metoda **pHorStrana** isto je kao i u roditeljskim klasama a telo metoda nije. Za drugi metod, **Kvadrat.pVerStrana**, ni zaglavlje metoda nije isto: umesto jednog argumenta sada su tri. Uopšte, zaglavlja izmenjenih metoda ne moraju se slagati. Ovaj metod takođe pokazuje kako se mogu pozivati istoimeni metodi iz „gornjih" klasa: dovoljno je navesti ime klase i razdvojiti ga tačkom od imena poruke.

Prva dva metoda klase **Kvadrat** u ovom primeru ukazuju na nove sintaksne mogućnosti. Treći metod, **Kvadrat.pObim**, pokazuje pravi razlog za redefinisanje metoda u nasledničkoj klasi: obimi pravougaonika i kvadrata izračunavaju se na različite načine. Nasleđivanje ima najviše smisla kada treba ovekovečiti male razlike između srodnih objekata.

Za kvadrat je potrebna samo jedna promenljiva a ne dve i bilo bi dobro da se nasleđivanjem mogu eliminisati nepotrebne promenljive iz prethodničkih klasa. To nije moguće, pa radi uštede memorije gore izloženu hijerarhiju možemo izvrnuti: na vrhu bi bila klasa **Kvadrat** sa samo jednom promenljivom, nju bi nasledila klasa **Pravougaonik** (ona bi dodala još jednu promenljivu) itd. Da *Turbo Pascal 5.5* ispunjava treći uslov – automatsko upravljanje memorijom, između ova dva pristupa bi razlika bila vrlo mala.

Klasu **Kvadrat** ilustrujemo sledećim programom:

```
program tp1;
uses j12, j13;
var p : pravougaonikObim;
    k : Kvadrat;
begin
   p.pHorStrana(3);
   k.pHorStrana(3);
   p.pVerStrana(4);
   k.pVerStrana(1, 2, 3);
   p.pObim;
   k.pObim;
end.
```

Iste poruke, **pHorStrana** i **pObim**, upućene su različitim objektima. Poruka **pVerStrana** se po zaglavlju razlikuje od klase do klase. Ista poruka može proizvesti različita dejstva u različitim klasama. To svojstvo objekata se naziva *polimorfizmom* i predstavlja ogromnu praktičnu prednost objektnog programiranja. Polimorfizam olakšava pisanje programa: paskal bez objekata zabranjuje da se isto ime koristi u raznim delovima programa, što nas primorava da izmišljamo različita imena za iste operacije. Druga prednost je još važnija: algoritmi mogu – konačno – postati nezavisni od podataka. Umesto da svaki program opteretimo velikim brojem CASE naredbi, dovoljno je da u svakoj klasi postoji metod istog imena. Jednom napisan (i preveden) algoritam u glavnom programu može zauvek ostati isti, a ako su promene neizbežne – dopisujemo novu klasu.

## VIRTUELNI METODI

```
program tp1;
uses Crt;
type
  GornjaKlasa = object
    procedure wr;
    procedure wrln;
    end;
procedure GornjaKlasa.wr;
begin
   write('Ovo je GornjaKlasa.wr ')
end;
procedure GornjaKlasa.wrln;
begin
   wr; writeln
end;
type
  DonjaKlasa = object(GornjaKlasa)
    end;
var gKlasa : GornjaKlasa;
   dKlasa : DonjaKlasa;
begin
   ClrScr;
   gKlasa.wrln;
   dKlasa.wrln;
end.
```

Gornji program definiše dve klase, nazvane **GornjaKlasa** i **DonjaKlasa**. Pomoću njih ukazujemo na problem do kojeg dolazi prilikom nasleđivanja i upotrebe polimorfnih metoda. **GornjaKlasa** ima samo dva metoda: **wr** piše poruku na ekranu a **wrln** poziva **wr** i dodaje znak za kraj reda. Ova dva metoda su po svemu slična naredbama **write** i **writeln**. Zahvaljujući nasleđivanju, metodi **wr** i **wrln** automatski važe i u klasi **DonjaKlasa**. Rezultat ovog programa su dva reda na vrhu ekrana:

```
Ovo je GornjaKlasa.wr
Ovo je GornjaKlasa.wr
```

Dovde nema problema – radi se o klasičnom nasleđivanju. Pretpostavimo da hoćemo da napišemo metode **DonjaKlasa.wr** i **DonjaKlasa.wrln** ali tako da se na ekranu pojavi neki drugi tekst. Izgleda logično dodati samo metod **DonjaKlasa.wr**. Program glasi ovako:

```
program tp1;
uses Crt
type
  GornjaKlasa = object
    procedure wr;
    procedure wrln;
    end;
procedure GornjaKlasa wr;
begin
    write('Ovo je GornjaKla
end;
procedure GornjaKlasa wrln
begin
    wr; writeln;
end;
type
  DonjaKlasa = object(Gornj
    procedure wr;
    end;
procedure DonjaKlasa wr
begin
    write('Ovo je DonjaKl
end

var gKlasa   GornjaKlasa
    dKlasa   DonjaKla
begin
    ClrScr;
    gKlasa wrln
    dKlasa wrln
end
```

Sve izgleda logično: metod **dKlasa wrln** nije definisan ali nema ni potrebe jer će se izvršiti metod **GornjeKlasa.wrln**. On poziva metode **wr** i **writeln** pošto je metod **wr** sada definisan u klasi **DonjaKlasa** na ekranu treba da se pojavi upravo poruka koju on šalje na ekran Međutim rezultat ovog programa biće potpuno isti kao i rezultat i prethodnog iz metode **GornjeKlasa wrln** ne izvršava se metod **DonjeKlasa wr** nego se izvršava **GornjaKlasa wr** Ispostavlja se da prevodilac već tokom prevođenja poveže pogrešnu verziju sa metodom **GornjaKlasa wr** Metod **DonjaKlasa wr** postoji u programu ali će ga prevodilac uvek ignorisati Mora postojati način da se prevodiocu naredi da metode ne povezuje prerano, tj pre nego se sa sigurnošću može odrediti koji metod treba izvršiti U *Turbo Pascal* u 5 5 ovaj problem se rešava uvođenjem triju novih rezervisanih reči **virtual, constructor** i **destructor** Evo ispravne verzije programa.

```
    procedure wr
    end
procedure GornjaKlasa wr;
begin
    write('Ove je GornjaKlasa
end;
procedure GornjaKlasa.wrln,
begin
    wr; writeln;
end;
type
  DonjaKlasa = object(GornjaKlas
    procedure wr, virtual
    procedure wrln
    constructor init,
    destructor done;
    end;
procedure DonjaKlasa wr,
begin
    write| Ove je DonjaKlasa
end;
procedure DonjaKlasa wrln
begin
    wr  writeln
end;
constructor DonjaKlasa init
begin
end;
destructor DonjaKlasa done
begin
end
var gKlasa   GornjaKlasa
    dKlasa   DonjaKlasa
begin
    
    gKlasa wrln
    dK    init
    d       rln;
```

Sada je metod **wr** u klasi **DonjaKlasa** virtuelan iza zaglavlja metoda dodaje se rezervisana reč **virtual** Dodali smo metod **DonjaKlasa wrln** kao i metoda **init** i **done** Umesto rezervisanih reči **procedure** i **function** za njih se koriste službene reči **constructor** i **destructor** Metodi označeni sa te dve službene reči moraju postojati ako se u objektu koriste virtuelni metodi. Konstruktorom ćemo nazivati metod deklarisan službenom rečju **constructor**, a pod destruktorom ćemo podrazumevati metod deklarisan službenom rečju **destructor**

Prevodilac određuje adrese običnih metode tokom prevođenja. Takve adrese ostaju iste tokom izvršenja programa pa se radi o statičkim metodima Adrese virtuelnih metoda određuju se tokom izvršenja programa nisu poznate unapred pa takvo povezivanje nazivamo kasnim ili dinamičkim povezivanjem Turbo Pascal pravi tzv tabelu virtuelnih metoda (VMT) u kojoj su adrese metoda Kada se tokom izvršavanja programa naiđe na virtuelni metod prvo se pregleda tabela virtuelnih metode ne bi li se našla ispravna adresa. Postupak je sličan CASE naredbi Zato polimorfizam praktično eliminiše potrebu za CASE naredbama u tekstu programa i izbor premešta u tabelu virtuelnih metoda Umesto da u programu postoji npr sledeći izbor

možemo definisati klasu **Oblik** i njene podklase **Tacka, Krug, Kvadrat, Pravougaonik** i **Krive** U svakoj od tih klasa treba napisati metod **Crtanje** pa će umesto gornje CASE naredbe biti dovoljno samo poslati poruku **Tacka Crtanje**, **Oblik.Crtanje**, **Krug Crtanje** itd Time se postiže sledeće

1) Broj CASE naredbi svodi se na minimum.

2) Možemo dodavati nove tipove podataka bez dodatnog prevođenja programe – pa čak i bez originalnog teksta programe!

CASE naredba ruši načela modularnosti i apstrakcije podataka. Kad uvedemo novi tip podataka u program, moramo

ručno izmeniti sve CASE naredbe od interesa i iznova prevesti sve delove programa u kojima se one pojavljuju. To nije uvek moguće. Tekst programa najčešće nije pri ruci (ili nikad nije ni bio isporučen). Treba imati istu ili saglasnu verziju prevodioca sa kojom je preveden originalni programa i sl. Virtuelni metodi automatizuju CASE naredbu: dovoljno je deklarisati metod kao virtuelan i svi njegovi potomci moći će da ga koriste, blokiraju ili konstruktivno prerade. U *Turbo Pascal*-u 5.5 sasvim je moguće definisati virtuelnu podklasu tako da bez ikakvih problema radi sa TPU datotekom prevedenom pre godinu dana ili ko zna kada.

## PRAKTIČAN ZNAČAJ VIRTUELNIH METODA

Praktičan značaj ovakve mogućnosti je ogroman. Pretpostavimo da ste na *Turbo Pascal*-u 5.5 napisali program za materijalno poslovanje svog preduzeća i da ste tu obradili, recimo, sastavnice. Posle izvesnog vremena shvatite da se rastavnice vrlo malo razlikuju od sastavnica. Da bi ste ih isprogramirali treba samo da rastavnice napravite kao podklasu sastavnica i dodate onoliko metoda za koliko se u stvarnosti rastavnice razlikuju od sastavnica. Još jednostavniji primer je promena konstanti u programu, na primer izdvajanja iz ličnog dohotka ili porez na promet. Za svaku novu promenu treba ih samo redefinisati kao novu podklasu, izmeniti konstante u programskom tekstu i prevesti samo nove naredbe.

U običnim jezicima bilo bi potrebno prevesti i povezati iznova ceo program. U *Turbo Pascal*-u 5.5 izmeni dela specifikacije programa odgovara samo izmena dela teksta programa. Dovoljno je imati:

- deklaracije klasa iz **Interface** delova jedinica, uz
- TPU datoteke tih istih jedinica.

Objektno orijentisani programi dozvoljavaju *korisnicima* da menjaju i održavaju program nezavisno od originalnog programera. Što je u komercijalnoj eksploataciji najvažnije svojstvo programa. Ovakva mogućnost je ključna praktična prednost za softversku kuću i pisce programa. Time se nudi novi kvalitet – apsolutna fleksibilnost krajnjeg korisnika. Kasno povezivanje dozvoljava da se novi metodi i objekti bez problema povežu sa isporučenom bibliotekom programa. Jedina cena koja se plaća je kratka poseta tabeli virtuelnih metoda. Ta tabela se nalazi u segmentu podataka i svaki novi primerak objekta se vezuje na nju. Statički metodi se zovu direktno, a virtuelni se prevode kroz tabelu virtuelnih metoda. Pregled virtuelne tabele je veoma brz, ali ipak za nijansu sporiji nego pozivanje statičkih metoda. Ako u objektu uopšte ne postoje virtuelni metodi, prevodilac ni ne generiše odgovarajuću tabelu, pa se ne zauzima prostor u segmentu podataka i ne troši se vreme na pregled tabele.

Razlika između statičkih i virtuelnih metoda je razlika između odlučivanja tokom prevođenja i odlučivanja tokom izvršenja programa. Statički metod podrazumeva da programer tačno zna šta hoće i da će to uvek važiti tokom izvršenja programa. Virtuelni metod se svodi na priznanje da se tokom prevođenja ne zna šta će se događati, ali da će sam program moći to da dokuči dok se izvršava. Suština je u tome da izvestan deo informacija o tome koga treba pozvati tokom prevođenja nije dostupan. Zato se ta odluka mora odložiti sve dok određeni objekat u programu ne zaživi. Onda program interpretira objekat i automatski saznaje koji metodi se na njega mogu primeniti a koji ne.

Jasno je da uvek treba koristiti metode, a ne obične procedure ili funkcije. Prednosti objektnog programiranja su prevelike a da se ne bi koristile. No, tu se zahteva jedna druga vrsta razmišljanja. Treba se odlučiti da li će metod biti statički ili virtuelan. Statički metodi se brže izvršavaju i mogu menjati deklaraciju kroz hijerarhiju. Samo virtuelni metodi nude toliko željenu fleksibilnost. Ako programer ne može u trenutku pisanja programa da zamisli zašto bi neko želeo da promeni ono što je on napisao, to uopšte ne znači da takva potreba neće kasnije iskrsnuti. U principu, uvek je bolje da metod bude virtuelan nego statički.

## TABELA VIRTUELNIH METODA

Tabelu virtuelnih metoda automatski pravi sam prevodilac, a program je direktno ne menja. Pokazivači na tu tabelu se konstruktorima smeštaju direktno u objekte tako da program ni njih ne menja direktno. Prva reč u tabeli virtuelnih metoda sadrži dužinu primeraka pridruženog tipa objekta. Tu informaciju koriste konstruktori i destruktori da bi znali koliko bajtova da pridodele ili odbace uz pomoć proširene sintakse naredbi new i dispose.

Druga reč u tabeli virtuelnih metoda sadrži negativnu dužinu primeraka pridruženog tipa objekta. Na osnovu sadržaja ove reči otkriva se da li je objekat pripremljen za rad (tj. da li je constructor bio pozvan) a takođe se proverava konzistentnost tabele. Provera virtuelnih poziva omogućava se prevodilačkom direktivom $R+. Tada prevodilac generiše odgovarajuću proveru pre svakog poziva virtuelnog metoda. Proverava se da li je prva reč tabele različita od 0, da li je suma prve i druge reči jednaka 0. Ako se ustanovi greška, prikazuje se poruka sa rednim brojem 210.

Uključivanje provere čini programe nešto većim i sporijim, a ostavlja se na volju programeru da proveru ostavi i u finalnoj verziji programa.

Sa rednim brojem četiri u tabeli virtuelnih metoda počinje spisak 32 bitnih pokazivača na metode, po jedan na svaki virtuelni metod u objektu, po redosledu deklarisanja. To su adrese ulaznih tačaka u virtuelne metode.

Konstruktori objekata sadrže specijalne grupe mašinskih naredbi za smeštanje početka tabele virtuelnih metoda u objekat koji se priprema za rad. Priprema se odigrava automatski tako da je polje **self** u tabeli virtuelnih metoda već postavljeno kada se izvrši naredba **begin** u konstruktoru. Zato konstruktor može zvati virtuelne metode, iako sam ne može biti virtuelan metod.

Uobičajeno ime za konstruktor je **init**, mada možemo zadati i bilo koje drugo ime. Ne sme postojati više od jednog konstruktora po klasi. Bilo koji objekat koji deklariše ili nasleđuje makar jedan virtuelni metod mora takođe deklarisati ili naslediti bar jedan konstruktor.

Ako klasa sadrži virtuelne metode i pošaljemo joj neku poruku a da prethodno nismo izvršili konstruktor, rezultat je neizvestan. Najverovatnije ćemo morati da isključimo računar ili nam sleduje problem neke druge vrste.

Statički metodi mogu menjati deklaracije od klase do klase. To za virtuelne metode ne važi:

– kada se metod jednom proglasi za virtuelan, mora ostati takav u naslednićkim klasama i
– deklaracije virtuelnog metoda se ne može menjati u naslednićkim klasama. Nije moguće pisati virtuelne metode koji bi primali različit broj argumenata, kao što to mogu, npr. standardne naredbe paskala write i writeln.

# NAREDBE new I dispose

Sintaksa naredbi **new** i **dispose** proširuje se i na virtuelne metode. Dinamički objekat koji sadrži virtuelne metode mora prvo pozvati svoj konstruktor:

```
var
   PKrug   ^Krug
   PKrug^ Init(600, 100  30)
```

Zatim se poruke mogu slati na uobičajeni način korišćenjem imena pokazivača i strelice umesto uvođenja imena objekta:

```
StariPolozaj  = PKrug^ DajX;
```

Procedure **new** može imati dva parametra – ime pokazivača kao prvi i poziv konstruktora kao drugi parametar:

```
New(PKrug,Init(600, 100, 30))
```

Te naredbe se izvršava kao da su bile napisane dve naredbe

```
New(PKrug)
PKrug^ Init(600  100  30)
```

Konstruktor **Init** dinamički zauzima memoriju korišćenjem specijalnog koda koji se automatski generiše tokom prevođenja konstruktora. Ime objekta ne može doći ispred **Init**, jer se zove procedura **new**, a objekat kojeg pripremamo za rad sa Init još ne postoji. Prevodilac pronalazi ispravan Init metod pomoću tipa pokazivača koji se uvodi kao prvi parametar.

Funkcijski oblik procedure **new** takođe prihvata **constructor** objekta kao drugi parametar

```
PLuk := New(LukPok  Init(600  100, 25  0
90));
```

Objekte iz dinamičke memorije brišemo naredbom **dispose**

```
Dispose(PKrug),
```

Objekat može sadržati pokazivače na dinamičke strukture ili objekte koji se moraju brisati u određenom redosledu. Sve takve radnje za brisanje dinamičkog objekta treba prikupiti u jedan ili više destruktora. Destruktor kombinuje brisanje iz dinamičke memorije sa svim drugim radnjama koje su neophodne za brisanje objekta. Destruktori se mogu nasleđivati, a mogu biti statički ili virtuelni. Pošto različiti objekti sasvim sigurno zahtevaju različite procedure brisanja i gašenja, svi destruktori treba da budu virtuelni. Destruktori treba da budu uparani sa konstruktorima.

Rezervisana reč **destructor** nije neophodna baš za svaki metod u kome se nešto briše. Prava snaga destruktora vidi se tek kada treba brisati polimorfne objekte iz dinamičke memorije (*heap*). Prava veličina objekta se ne može saznati tokom prevođenja. **destructor** prima kopiju parametra **self** na steku u trenutku poziva i iz tabele virtuelnih metoda očitava tačnu dužinu objekta. Time poziv destruktora garantuje da će se ispravan broj bajtova osloboditi iz dinamičke memorije. Nikakva šteta, međutim, ne može nastati ako se destruktori primene na statičke objekte.

Objekat može biti polimorfan kada se program prevodi ali nikad nije polimorfan kada se program izvršava.

Sintaksno gledano, **destructor** se mora pozvati kroz proširenu naredbu **Dispose**, npr:

```
Dispose(TackaPok Done)
```

To je isto kao da su izvršene sledeće dve naredbe

```
TackaPok^ Done;
Dispose(TackaPok)
```

Poziv destruktora van procedure **Dispose** ništa ne briše sam po sebi. Sam destruktor može biti i bez naredbi, ali ga treba navoditi radi tabele virtuelnih metoda.

# VIŠESTRUKO I PONOVLJENO NASLEĐIVANJE

*Turbo Pascal 5.5* ne ispunjava sedmi uslov – višestruko i ponovljeno nasleđivanje. Višestruko nasleđivanje možemo simulirati definisanjem po jedne promenljive za svaku klasu koju treba naslediti. Primera radi:

```
type
   Prozor   object
      zaglavlje   Pravougaonik
      ekran : Multiprocesor;
      end;
```

Ovaj primer pretpostavlja postojanje klasa **Pravougaonik** i **Multiprocesor**.

# USAGLAŠENOST TIPOVA

Nasleđivanje donekle menja usaglašenost tipova u *Turbo Pascal*-u. Naslednička klasa nasleđuje i usaglašenost tipa sa svojim prethodnicima. Takva usaglašenost može biti između

- primeraka objekata,
- pokazivača na primerke objekta, i
- formalnih i stvarnih parametara.

Usaglašenost tipova ide samo od roditeljske klase ka nasledniku. Naslednički tipovi se mogu sasvim slobodno koristiti umesto roditeljskih tipova, ali obratno nije moguće.

Razmotrimo sledeće deklaracije

```
type
   MestoPok    = ^Mesto;
   TackaPok    = ^Tacka;
   KrugPok     = ^Krug,
var
   AMesto      : Mesto,
   JednaTacka  : Tacka;
   AKrug       : Krug;
   PMesto      : MestoPok;
   PTacka      : TackaPok;
   PKrug       : KrugPok;
```

U prisustvu tih deklaracija sledeća pridodeljivanja su legalna

```
AMesto     := JednaTacka;
JednaTacka:= AKrug;
AMesto     := AKrug.
```

Obratna pridodeljivanja nisu dozvoljena.

Ovaj koncept je potpuno nov za paskal. Naslednički tip je ili tačno iste dužine kao i roditeljski ili, što je obično slučaj, je veći. Jedno je sigurno – nikako ne može biti manji. Pridodeljivanje prethodničke klase nasledničkoj klasi ne bi postavilo sve polje u primerku nasledničke klase i zato je zabranjeno. Pridodeljivanjem se kopiraju samo zajednička polje roditelj

skog i nasledničkog tipa čak i ako to nije sasvim očigledno iz same naredbe pridodeljivanja.

Usaglašenost tipova takođe važi za pokazivače na objektima, pod istim opštim uslovima kao i za druge tipove. Pokazivači na naslednike mogu se pridodeliti pokazivačima na roditelje. Za prethodno navedene deklaracije sledeća pridodeljivanja pokazivača bile bi legalna:

```
PTecka := PKrug;
PMesto := PTecka;
PMesto := PKrug;
```

Obratna pridodeljivanja nisu dozvoljena.

Formalni parametar datog objekta može za svoj stvarni parametar prihvatiti ili objekat istog tipa ili bilo koji naslednički tip. Ako je data deklaracija procedure

```
procedura Vucenje(Cilj : Tacka)
```

stvarni parametri bi mogli biti tipa **Tecka** ili **Krug**, ali ne i tipa **Mesto**. Cilj bi takođe mogao biti VAR parametar.

Bez obzira na to, postoji drastične razlike između prenosa parametara sa i bez naredbe VAR. Ako je parametar deklarisan kao VAR, onda se prenosi pokazivač na stvarni objekat. Ako ne, onda se prenosom po vrednosti samo kopira stvarni parametar u neimenovanu lokalnu varijablu procedure. Ta kopija sadrži samo polje i metode koje sadrži tip formalnog parametra. Dakle, stvarni parametar se bukvalno prenosi na tip formalnog parametra.

Slično, ako je formalni parametar pokazivač na objekat, stvarni parametar može biti pokazivač na taj objekat ili na bilo koji naslednički objekat. Za sledeću deklaraciju procedure

```
procedura Slika Dodavanje(NovaSlika : TackaPok)
```

stvarni parametri bi mogli biti tipa **TackaPok** ili **KrugPok**, ali ne i tipa **MestoPok**.

Objekat ima fiksan broj elemenata. Element objekta je ili

- polje, tj. sadrži podatke nekog tipa, ili
- metod, tj. izvodi operacije nad objektom.

Slično deklaraciji varijable, deklaracija polja zadaje tip polja i ime. Slično deklaraciji procedure ili funkcije, deklaracija metoda zadaje zaglavlja procedure funkcija, **constructora** ili **destructora**.

Klasa (tip objekta) može naslediti elemente neke druge klase. Ako T2 nasleđuje T1, onda je T2 naslednik ili potomak od T1, a T1 je roditelj ili predak za T2.

Nasleđivanje je tranzitivno, pa ako T3 nasleđuje T2 i T2 nasleđuje od T1, onda T3 takođe nasleđuje T1. Domen objekta obuhvata i sve potomke.

Za razliku od drugih tipova, klasa se može deklarisati samo u naredbi **type**. Tip objekta ne može biti deklarisan u naredbi var, a ni u okviru procedure, funkcije ili bloka unutar metoda.

Klasa ne može biti element datotečnog tipa, niti to može biti bilo koji strukturirani tip koji sadrži klasu.

Oblast važenja imena širi se kroz domen klasa i obuhvata sve procedure, funkcije, **constructore** i **destructore** koji sadrže metode klase i njene potomke. Imena elemenata klase moraju biti jedinstvena u objektu, u njegovim potomcima i u svim njihovim metodama.

Deklaracija metoda u tipu objekta odgovara **forward** deklaraciji tog metoda. U okviru deklaracije klase zaglavlja metoda može zadavati parametre za objekat iako njegova deklaracija nije dovršena.

Za pridodeljivanje objekata važe sledeća pravila:

– objekat tipa T2 je usaglašen u odnosu na pridodeljivanje sa tipom objekta T1 ako T2 pripada domenu od T1.

– pokazivač tipa T2 koji pokazuje na tip objekta T2 je usaglašen u odnosu na pridodeljivanje sa pokazivačem tipa T1 koji pokazuje na objekat tipa T1, ako je T2 u domenu od T1.

Logična pravila važe i za prenos parametara u potprograme. Ako je formalni parametar tipa T1 onda stvarni parametar može biti tipa T2, pod uslovom da je T2 u domenu od T1.

Interni oblik podataka u objektima podseća na slog. Polja objekta smeštaju se po redosledu deklarisanja kao neprekidan niz promenljivih. Sva polja nasleđena od predaka smeštaju se pre novih polja definisanih u potomcima.

Ako objekat definiše virtuelne metode, konstruktore ili destruktore, prevodilac stvara i 16-bitno polje tzv. polje tabele virtuelnih metoda. Ono sadrži adresu tabele virtuelnih metoda datog objekta u segmentu podataka.

Polje tabele virtuelnih metoda počinje odmah iza običnih polja u objektu. Kada objekat nasleđuje virtuelne metode, konstruktore ili destruktore, istovremeno nasleđuje i polje tabele virtuelnih tabela, tako da se nova takva polje u potomcima ne stvaraju.

Samo konstruktor (jedan ili više njih) menja polje tabele virtuelnih metoda. Sam program nikada eksplicitno ne analizira ili menja to polje.

Deklaracija konstantnog tipa objekta ima istu sintaksu kao i deklaracija konstantnog sloga. Nikakva vrednost se ne može zadati za elemente metode. Konstantni oblici koji sadrže virtuelne metode ne smeju se postavljati pozivom konstruktora. Priprema za rad sprovodi prevodilac automatski.

Adresni operator @ primenjen na kvalifikovani metod vraća pokazivač na ulaznu tačku u metod.

Objekat se može pridodati sopstvenom potomku. To je projekcija potomka u prostor pretka. Takvim pridodeljivanjem se objekat ne priprema automatski za rad.

Funkcija **TypeOf** poredi tipove objekata. Ona vraća pokazivač na tabelu virtuelnih metoda tog objekta. Deklariše se ovako

Ulazni podatak je ime ili primerak objekta. U oba slučaja rezultat je pokazivač na tabelu virtuelnih metoda tog objekta. **TypeOf** se može primeniti samo na objekte koji imaju tabelu virtuelnih metoda, inače će biti prijavljena greška.

Dosadašnji primeri su služili samo da se ilustruju osnovni koncepti objektnog programiranja u *Turbo Pascal*-u 5.5. Na primer, objekat Pravougaonik bi trebalo dopuniti koordinatama za grafički prikaz na ekranu i sl. Sada navodimo primer nasleđivanja u znatno realnijoj situaciji. Radi se o sortiranju redova teksta.

Pretraživanje podataka uz pomoć računara je efikasno samo ako su podaci sortirani, tj. uređeni po nekom kriterijumu. Smatra se da se čak 30% svih operacija koje računari izvode može podvesti pod sortiranje. Postoji mnogo algoritama za sortiranje i nema najefikasnijeg u nekom apsolutnom smislu. Naravno, za date uslove postoje dobri i bolji algoritmi, a dva opštepriznata algoritma su Šel sort (po imenu autora, *Shell*) i brzi sort (*quick sort*). Brzi sort je jedan od najbržih opštih sortova, ali u najgorem slučaju ne daje najbolje rezultate.

Suština svakog algoritma za sortiranje svodi se na tri elementa:

– izbor elemenata za poređenje,
– poređenje elemenata, i
– premeštanje elemenata u nizu.

Po prvom elementu, strategiji izbora, sortovi se međusobno razlikuju. Drugi i treći element u principu nisu sporni, tj. trebalo bi da budu samo tehničke prirode. U njima se ogleda priroda podataka koji se uređuju, tako da program za sortiranje direktno zavisi od oblika podataka. Najčešći slučaj je uređiva-

nje nekih tekstualnih jedinica – reči, rečenica, pasusa i slično. Drugi čest slučaj je uređivanje brojeva. Bitna je razlika između sortiranja teksta i sortiranja brojeva, jer je tekst po pravilu mnogo duži od broja. Operacije poređenja i premeštanje teksta bile bi mnogo sporije ako bi se prilikom svakog poređenja i premeštanja on fizički premeštao po memoriji ili disku. Zato se umesto sortiranja teksta uvodi indeksiranje: svakom tekstualnom elementu pridodeljuje se pokazivač (indeks) kojim se određuje njegovo mesto u nizu. Umesto premeštanja teksta menja se pokazivač. Sortirani tekst se onda čita po redosledu pokazivača (a oni se čitaju linearno, npr. for petljom). Tako se mnogo dobija na vremenu, a nešto se gubi na prostoru.

Sortiranje prostih nizova je lako, jer je element istovremeno i indeks. Za sortiranje teksta ili, u opštem slučaju, slogova, moraju se ustanoviti dodatni ključevi za sortiranje. Jedinica **Linije0** sadrži dva objekta koji su u suštini nizovi tekstualnih linija. Zovu se **Linije** i **Linije2**. Konstante i varijable su:

| | |
|---|---|
| **max_duzine_podataka** | – maksimalan broj linija teksta, |
| **sirine** | – širina sortnog ključa |
| **tip_podataka** | – tekst date širine, |
| **raspon** | – domen za **niz_podataka** i **niz_kljuceva**, |
| **slog_podataka** | – zajednica podataka i ključeva |
| **slog_za_sort** | – sve što treba da se sortira smešteno u jedan slog, da bi se elegantnije prenelo proceduri za sortiranje |

Objekat **Linije** definišemo na tipičan način: samo jedna promenljiva, **lin**, i niz metoda koji nad njom dejstvuju. Prve dve procedure, **init** i **done**, su očigledne. U ovom slučaju u njima se baš ništa ne dešava, ali ih ipak (zbog VMT tabele) treba uredno navoditi na početku i na kraju programa. **sort** je upravo procedura čije nasleđivanje želimo da ilustrujemo. U roditeljskom objektu, **Linije.sort** zove proceduru **SelSort**, u kojoj se pravo sortiranje i događa. **SelSort** se ne navodi u definiciji objekta, pa je to privatna procedura za **Linije**. Linije sortiramo porukom **sort**. To je ne samo lakše za pamćenje, nego ima i tu dodatnu prednost da možemo napraviti podklasu klase **Linije**, nazovimo je **Linije2**, i da u njoj pod imenom **Sort** zapravo izvršimo neki sasvim drugi algoritam za sortiranje. Taj objekat, **Linije2**, definisan je odmah iza objekta **Linije**, ali još uvek pre službene reči **implementation**. **Linije2** ima samo tri metoda: **init**, **done** i **sort**. U sintaksnom pogledu treba naglasiti da je u objektu **Linije init** bila procedura, a u **Linije2** je **constructor**. Slično, **Linije.done** je procedura, a **Linije2.done** je destruktor. Konačno, objekat **Linije2** postoji samo da bi izmenio način sortiranja, tako da je procedura **sort** u njemu deklarisana kao **virtual**. **Linije2.sort** zove proceduru **rekurzivni_brzi_sort**, a **Linije.sort** zove **SelSort**, i to je jedina razlika. U principu, možemo dodati i neki treći algoritam sortiranja. Dovoljno je imati interface deo jedinice **Linije0** i datoteku LINIJE0.TPU pa da se sortiranje može prepraviti i bez teksta procedure **Linije.sort**.

Preostali metodi i funkcije uglavnom postavljaju ili očitavaju vrednosti promenljivih iz objekta.

```
{$M 65520,0,655360}
unit Linije0
INTERFACE
const max_duzina_podataka = 200;
      sirina = 20;

type
  tip_podataka = string[sirina];
  raspon = 1..max_duzina_podataka;
  niz_podataka = array [raspon] of tip_poda-
taka;
  tip_kljuceva = tip_podataka;
  niz_kljuceva = array [raspon] of tip_klju-
ceva;
  slog_podataka = record
    podaci : tip_podataka;
    kljuc : tip_kljuceva;
    end;
  niz_slogova = array [raspon] of slog_poda-
taka;
  niz_indeksa = array [raspon] of raspon;
  slog_za_sort = record
    broj_podataka : word;
    podaci : niz_slogova;
    indeks_za_sort : niz_indeksa;
    rastuci : boolean;
    end;

Linije = object
  lin : slog_za_sort;
  procedure init;
  procedure done;
  procedure sort;
  procedure wr;
  procedure wrln;
  procedure postavi_podatke(podaci0 : niz_po-
dataka);
  procedure postavi_kljuceve(kljucevi :
niz_kljuceva);
  procedure postavi_broj_linija(n : integer);
  function broj_linija: integer;
  procedure postavi_rastuci(b : boolean);
  function rastuci : boolean;
  end; { Linije }

Linije2 = object(Linije)
  constructor init;
  destructor done;
  procedure sort; virtual;
end; { Linije2 }

IMPLEMENTATION

procedure selsort(var p : slog_za_sort);
var i, j, e, m, skip : word;
    uredu : boolean;

begin
  with p do begin
    if broj_podataka = 0 then exit
    else if broj_podataka > max_duzina_poda-
taka then
            broj_podataka := max_duzina_poda-
taka;
    for i := 1 to broj_podataka do
        indeks_za_sort[i] := i;
    skip := broj_podataka;
    while skip > 1 do begin
      skip := skip div 2;
      repeat
        uredu := true;
        for j := 1 to broj_podataka - skip
do begin
          i := j + skip;
          e := indeks_za_sort[i];
          m := indeks_za_sort[j];
          if podaci[e].kljuc <
podaci[m].kljuc then begin
            { premestanje }
            indeks_za_sort[i] := m;
            indeks_za_sort[j] := e;
            uredu := false
          end; { if }
        end; { for }
      until uredu;
    end; { while skip > 1 }
  end; { with p }
end; { shell sort }

procedure rekurzivni_brzi_sort(var p :
slog_za_sort );
var k: word;
```

```
    srednji, privremeno : slog_podataka;

procedure sort(levi, desni : word);
var i, j, privremeno : word,
    srednji : slog_podataka;
begin
  with p do begin
    i := levi;
    j := desni;
    srednji := podaci[ indeks_za_sort[ (levi
+ desni) div 2] ];
    repeat
      while podaci[ indeks_za_sort[i] ]
.kljuc < srednji.kljuc do inc(i);
      while srednji.kljuc < podaci[ in-
deks_za_sort[j] ].kljuc do dec(j);
      if i <= j then begin
        privremeno := indeks_za_sort[i];
        indeks_za_sort[i] :=
indeks_za_sort[j];
        indeks_za_sort[j] := privremeno;
        inc(i);
        dec(j);
      end; { if }
    until i > j;
    if levi < j then sort(levi,j);
    if i < desni then sort(i,desni);
  end; { with }
end; { sort }

begin
     if p.broj_podataka = 0 then exit
     else if p.broj_podataka >
max_duzina_podataka then
               p.broj_podataka := max_duzi-
na_podataka;
     { priprema indeksa }
     for k := 1 to p.broj_podataka do
        p.indeks_za_sort[k] := k;
     sort(1,p.broj_podataka)
end; { rekurzivni_brzi_sort }

procedure inverzija(var p : slog_za_sort);
var prvi, poslednji, privremeno : word;

begin
  with p do begin
    if broj_podataka = 0 then exit
    else if broj_podataka > max_duzina_poda-
taka then
              broj_podataka := max_duzina_poda-
taka;
    prvi := 1;
    poslednji := broj_podataka;
    while prvi < poslednji do begin
      privremeno := indeks_za_sort[prvi];
      indeks_za_sort[prvi] := in-
deks_za_sort[poslednji];
      indeks_za_sort[poslednji] :=
privremeno;
      inc(prvi);
      dec(poslednji);
      end; { while }
  end; { with p }
end; { reverse_array_order }

procedure Linije.init;
begin
end;

procedure Linije.done;
begin
end;

procedure Linije.sort;
begin
     selsort(lin);
     if not lin.rastuci
   then inverzija(lin);
end; { Linije.sort }

procedure Linije.wr;
var i, j : integer;
begin
     for i := 1 to lin.broj_podataka do begin
       j := lin.indeks_za_sort[i];
       write(lin.podaci[j].podaci);
       end;
end;

procedure Linije.wrln;
var i, j : integer
begin
     for i := 1 to lin.broj_podataka do begin
       j := lin.indeks_za_sort[i];
       writeln(lin.podaci[j].podaci);
       end;
end;

procedure Linije.postavi_kljuceve;
var i : integer;
begin
     for i := 1 to lin.broj_podataka do
       lin.podaci[i].kljuc := kljucevi[i];
end;

procedure Linije.postavi_podatke;
var i : integer;
begin
     for i := 1 to lin.broj_podataka do
       lin.podaci[i].podaci := podaci0[i];
end;

procedure Linije.postavi_broj_linija;
begin
     lin.broj_podataka := n;
end;

function Linije.broj_linija;
begin
     broj_linija:= lin.broj_podataka;
end;

procedure Linije.postavi_rastuci;
begin
     lin.rastuci := b;
end;

function Linije.rastuci;
begin
     rastuci := lin.rastuci;
end;

constructor Linije2.init;
begin
end;

destructor Linije2.done;
begin
end;

procedure Linije2.sort;
begin

rekurzivni_brzi_sort(lin);
     if not lin.rastuci
   then inverzija(lin);
end;
end. { Linije 2}
```

## Test_Lin – GLAVNI PROGRAM ZA SORTIRANJE

Glavni program kojim se **Linija** i **Linija2** testiraju i demonstriraju je **Test_Lin**. On preuzima potrebne tipove iz jedinica **Linije0** i u njih postavlja neke vrednosti. Varijable **linija** i **linija2** su objekti (primerci) klasa **Linija** i **Linija2**. Primetimo da je upućivanje poruka tim objektima identično. Na primer naredbe **linija.sort** i **linija2.sort** proizvode potpuno isti efekat – spolja gledano. To je klasičan primer polimorfizma. Dodatna prednost za programera je što više ne mora da izmišlja slična ali i pomalo različita imena za ista operacija. Sort je uvek sort i zašto se sortiranja svakom objektu ne bi naređivalo na isti način?

```
[$M 65520,0,655360]
program test_lin;
uses crt, linije0;
{$V-}
var  podaci : niz_podataka,
     kljuc : niz_kljuceva,
     linija   Linija;
     linija2 : Linija2;
     i, j : integer;

{ test_lin }
begin
      clrscr;
      linija init;
      writeln('sortiranje niza linija - pri-
mer za nasle'ivanje ');
      writeln,
      podaci[1]  = '3', podaci[2] := '2'; po-
daci[3] := '1';
      podaci[4]  = '6'; podaci[5] := '4'; po-
daci[6]  = '5';
      linija postavi_broj_linija(6);
      writeln('postavljeni broj linija ', li-
nija broj_linija);
      writeln;

      linija postavi_podatke(podaci),
      for i := 1 to sirina do
         kljuc[i]  = podaci[i],
      linija postavi_kljuceve(kljuc),
      writeln( podaci sortirani saviae '),
      linija postavi_rastuci(true),
      linija sort;
      linija wr;
      { obratno }
      linija.postavi_rastuci(false);
      linija sort,

      writeln,
      writeln( podaci sortirani sanize )
      linija wr;
      writeln, writeln; writeln('linija2 -
nasle"ivanje ');
      linija2 init,
      linija2 lin := linija lin;
      linija2 wr,
      writeln,

      linija2 postavi_rastuci(true);
      linija2 sort,
      writeln; writeln('sortiranje na drugi
nacin '),
      linija2 wr,
      linija done, linija2 done;
end  { test_lin }
```

## MATRIČNI ŠTAMPAČI KAO OBJEKTI

Objektno orijentisano programiranje pruža elegantan način da se u program ugradi podrška za rad sa više štampača. Ograničimo se na matrične štampače jer za njih postoji dobro definisan standard firme Epson. Apstraktne klase na prirodan način opisuju standarde u nekoj oblasti. Na vrh hijerarhije štampača postavljamo apstraktnu klasu **EpsonStampac**. U njoj su samo pobrojane osobine štampača, tj. poruke koje mu želimo uputiti. Radi sintaksne ispravnosti, metodi moraju biti prisutni ali u njima nema naredbi. Evo klase **EpsonStampac**, iz jedinice EPSON.PAS

```
unit epson;
{ naredbe za stampanje na matricnom stampacu
Epson }
interface
type
    EpsonStampac = object
      procedure kompresovanaSlova,
      procedure neKompresovanaSlova;
      procedure prosirenaSlova,
      procedure neProsirenaSlova;
      procedure naglasenaSlova;
      procedure neNaglasenaSlova;
      procedure dvostrukaSlova;
      procedure jednostrukaSlova,
      procedure eksponenti;
      procedure indeksi,
      procedure obicnaSlova;
      procedure podvlacenje,
      procedure bezPodvlacenja;
      procedure linijaNaStrani (brojLinija :
word);
      procedure sirinaStrane (sirina : word);
      procedure pripremaStampaca;
      procedure preskokPerforacije;
      procedure bezPreskokaPerforacije
      procedure vlasnLinije,
      procedure briseZnakove;
      end; { EpsonStampac }
implementation
procedure EpsonStampac kompresovanaSlova
begin
end;
procedure EpsonStampac neKompresovanaSlova,
begin
end;
procedure EpsonStampac prosirenaSlova,
begin
end,
procedure EpsonStampac.neProsirenaSlova,
begin
end,
procedure EpsonStampac naglasenaSlova;
begin
end;
procedure EpsonStampac neNaglasenaSlova;
begin
end,
procedure EpsonStampac dvostrukaSlova;
begin
end;
procedure EpsonStampac jednostrukaSlova;
begin
end,
procedure EpsonStampac eksponenti;
begin
```

```
end;
procedure EpsonStampac.indeksl;
begin
end;
procedure EpsonStampac.obicnaSlova;
begin
end;
procedure EpsonStampac.podvlacenje;
begin
end;
procedure EpsonStampac.bezPodvlacenja;
begin
end;
procedure EpsonStampac.linijaNaStrani (broj-
Linija : word);
begin
end;
procedure EpsonStampac.sirinaStrane (sirina
: word);
begin
end;
procedure EpsonStampac.pripremaStampaca;
begin
end;
procedure EpsonStampac.preskokPerforacije;
begin
end;
procedure EpsonStampac.bezPreskokePerforaci-
je;
begin
end;
procedure EpsonStampac.visinaLinije;
begin
end;
procedure EpsonStampac.brisiZnakove;
begin
end;
end. { epson }
```

Zatim definišemo klase koje ćemo koristiti u programu. Najrasprostranjeniji je model FX85 pa pravimo klasu FX85 u jedinici UNITFX85.PAS. U njoj je većina metoda „ispunjena naredbama".

```
unit UnitFX85;
{ naredbe za stampanje na matricnom stampacu
Epson FX85/105}
interface
uses Epson, Printer;
type
    FX85 = object(EpsonStampac)
      procedure wr(znakovi : string);
      procedure wrln(znakovi : string);
      procedure kompresovanaSlova;
      procedure neKompresovanaSlova;
      procedure prosirenaSlova;
      procedure neProsirenaSlova;
      procedure naglasenaSlova;
      procedure neNaglasenaSlova;
      procedure dvostrukaSlova;
      procedure jednostrukaSlova;
      procedure eksponentl;
      procedure indeksl;
      procedure obicnaSlova;
      procedure podvlacenje;
      procedure bezPodvlacenja;
      procedure pripremaStampaca;
      procedure preskokPerforacije;
      procedure bezPreskokePerforacije;
      procedure brisiZnakove;
      procedure pica;
      procedure elite;
      procedure lf;
      procedure ff;
      procedure cr;
    end; { FX85 }
implementation
procedure FX85.wr;
begin
    write(Lst, znakovi);
end;
procedure FX85.wrln;
begin
    writeln(Lst, znakovi);
end;
procedure FX85.kompresovanaSlova;
begin
    write(Lst, chr(15));
end;
procedure FX85.neKompresovanaSlova;
begin
    write(Lst, chr(18));
end;
procedure FX85.prosirenaSlova;
begin
    write(Lst, chr(18));
end;
procedure FX85.neProsirenaSlova;
begin
    write(Lst, chr(20));
end;
procedure FX85.naglasenaSlova;
begin
    write(Lst, chr(27), 'E');
end;
procedure FX85.neNaglasenaSlova;
begin
    write(Lst, chr(27), 'F');
end;
procedure FX85.dvostrukaSlova;
begin
    write(Lst, chr(27), 'G');
end;
procedure FX85.jednostrukaSlova;
begin
    write(Lst, chr(27), 'H');
end;
procedure FX85.eksponentl;
begin
    write(Lst, chr(27), chr(14));
end;
procedure FX85.indeksl;
begin
    write(Lst, chr(27), chr(15));
end;
procedure FX85.obicnaSlova;
begin
    write(Lst, chr(27), 'T');
end;
procedure FX85.podvlacenje;
begin
    write(Lst, chr(27), '-1');
end;
procedure FX85.bezPodvlacenja;
begin
ALISTING - write(Lst, chr(27), '-0')
end;
procedure FX85.pripremaStampaca;
begin
    write(Lst, chr(27) + chr(64))
end;
procedure FX85.preskokPerforacije;
begin
    write(Lst, chr(27), 'N');
end;
procedure FX85.bezPreskokePerforacije;
begin
    write(Lst, chr(27), 'O');
```

```
end
procedure FX85 briseZnakove;
begin
     write(Lst, chr(24));
end
procedure FX85 pica;
begin
     write(Lst, chr(27) + 'P');
end;
procedure FX85 elite;
begin
     write(Lst, chr(27) + 'M');
end;
procedure FX85 ff;
begin
     write(Lst, chr(12));
end;
procedure FX85 lf;
begin
     write(Lst, chr(10));
end;
procedure FX85.cr;
begin
     write(Lst, chr(13));
end
end  { Unit FX85}
```

Postoji više načina za slanje naredbi matričnom štampaču. ASCII simboli do broja 31 se ne mogu odštampati jer ih štampač interpretira kao naredbe. Njih možemo slati štampaču isključivo preko CHR() funkcije. Štampač FX85 na broj 13 reaguje prelaskom u novi red, pa u nizu write naredbi metoda cr pišemo chr(13). Broj 27 je poseban: iza njega dolazi jedan ili više znakova koje štampač sve interpretira kao jednu naredbu. Simbole posle **chr(27)** možemo pisati na tri načina:

```
chr(27), chr(77)
chr(27) + chr(77)
chr(27) + 'M'
```

Prvi način koristi zarez kao simbol za nastavak u okviru write ili writeln naredbe, drugi način sabira tj. prisajedinjuje znakove (rezultat funkcije **chr** je tipa **char**), a u trećem načinu je piše i eksplicitno. Sva tri načina pisanja su ekvivalentna za ASCII simbole veće od 31.

Velika popularnost štampača FX85 proizvela je veliki broj klonova – štampača koji isto reaguju na naredbe. Ponekad je ta usaglašenost delimična. Na primer, štampač *Logitec FT 5002* uglavnom radi kao klon Epsonovog standarda ali se u nekim pojedinostima razlikuje. Ta situacija savršeno se uklapa u termine objektnog programiranja: treba samo definisati potklasu klase FX85. Ona je u jedinici UNITLOGI.PAS.

```
procedure Logitec.eksponenti;
begin
     write(Lst, chr(27) + 'S' + c...);
end;
procedure Logitec.indeksi;
begin
     write(Lst, chr(27) + 'S' + c...);
end;
procedure Logitec obicnaSlova;
begin
     write(Lst, chr(27), 'T');
end;
end  { UnitLogi}
```

Razlika u naredbama za pisanje eksponenata i indeksa (slova podignutih ili spuštenih u odnosu na osnovu reda) je vrlo jednostavno opisana odgovarajućim statičkim metodima klase **Logitec**. Sledeći program pokazuje kako se naredbama klasa **FX85** i **Logitec** može upravljati štampačem *Logitec*.

```
program testlogi;
uses unitlogi;
var stampac : Logitec;
begin with stampac do begin
  wrln(''), kompresovanaSlova
    wrln('Kompresovana slova'); neKompresov
anaSlova;
  prosirenaSlova; wrln('Prosirena slova');
neProsirenaSlova;
  naglasenaSlova  wrln('Naglasena slova');
neNaglasenaSlova
  dvostrukaSlova; wrln('Dvostruka slova');
jednostrukaSlova;
  eksponenti  wrln('Eksponenti');  obicnaSlo
va;
  indeksi  wrln('Indeksi');  obicnaSlova
  podvlacenje  wrln('Podvlacenje');  bez pod
vlacenja
  p ipr   Sta p ca  wrln( prip   e
a );
  b   Znak vo  wrln( brise znakov  )
  ;     wrln(  )  wrln(  d        p
  ;
       wr m(   ;  wr n( Sada su   l
```

Odgovarajući program TESTFX85.PAS bi se razlikovao samo po korišćenju jedinica **unitfx85** i objekta **fx85** umesto jedinice **unitlogi** i objekta **Logitec**. Štampač treba da bude uključen prilikom aktiviranja ovog programa.

Klasa **EpsonStampac** je daleko od kompletnosti neophodne za komercijalnu upotrebu. Štampači reaguju na stotine naredbi pa bi odgovarajuća klasa bilo preterano dugačka za objavljivanje. Međutim, lako je dopuniti bilo koju od gornjih klasa, a poseban kvalitet u komercijalnoj prodaji programa je i to što bi korisnik – pod uslovom da ima *Turbo Pascal* – mogao sam da definiše nove osobine štampača.

Standardne jedinice proširuju *Turbo Pascal* u odnosu na standard paskala. Naporedo sa njima na disketama postoje mnoge klase ali su slabije dokumentovane. U datoteci OBJECTS.PAS nalazi se nekoliko osnovnih objekata. Njihov opis je u datoteci OOPDEMOS.DOC. Preko 12000 programskih redova nalazi se u datoteci OOPEMOS.ARC na disketi OOP/DEMOS/BGI/DOC. Postoji nekoliko velikih klasa, npr. deklaracija objekta **Spreadsheet** iz datoteke **TCSheet** dugačka je nekih 100 redova, sa oko 60 metoda. Neki objekti služe samo za programe u okviru kojih su nastali (ćelije u programu *Turbo Calc*, zid iz igre *Breakout*) a neki su opštiji (**Base**, **Stream**) i mogu se smatrati korisnim proširenjem *Turbo Pa*

scal-a kao jezika. Upoznavanje sa takvim klasama cilj je ovog poglavlja.

Hijerarhija isporučenih klasa je

```
Datoteka   Klasa

Objects    SProc = object
TCParser   ParserObj = object
TCInput    InputField = object
TCLStr     LString = object
Bounds     ObstacleList = object
TCRun      ProgramObject = object
TCMenu     Menu = object
TCSheet    Spreadsheet = object
TCScreen   Screen = object
             ScreenArea = object
             ColorTable = object
TCCell     Block = object
TCMenu     MenuItem = object
             MenuItemProc = object(MenuItem)
             MenuItemMenu = object(MenuItem)
Objects    List = object
ODemo      TestList = object(List)
           Location = object
             Cursor = object(Location)
               SaveScreen = object(Cursor)
Bricks     Block = object(Location)
               Ball = object(Block)
               Brick = object(Block)
Bounds     Obstacle = object(Location)
               Paddle = object(Obstacle)
               Boundary = object(Obstacle)
                 LeftBound = object(Boundary)
                 UpperBound = object(Boundary)
                 RightBound = object(Boundary)
                 LowerBound = object(Boundary)
Walls                              Wall = object(Obstacle)
Points                       Point = object (Location)
Figures      Circle = object (Point)
FigureDemo/
ListDemo         Arc = object (Circle)
Count      TextString = object(Location)
               Counter = object(TextString)
                 DownCounter = object(Counter)
                     LimitCounter = object(DownCounter)
TCUtil     HashTable = object
TCCell       CellHashTable = object(HashTable)
             FormatHashTable = object(HashTable)
             WidthHashTable = object(HashTable)
             OverwriteHashTable = object(HashTable)
Objects    Base = object
Forms        Form = object(Base)
Cards        CardList = object(Base)
Objects      Stream = object(Base)
                 DosStream = object(Stream)
TCCell           SSStream = object(DosStream)
Objects          BufStream = object(DosStream)
Forms              FStream = object(BufStream)
                       {FStream = object(Forms.FStream)}
Odemo              TestStream = object(BufStream)
Objects      Node = object(Base)
Forms          Field = object(Node)
                   FText = object(Field)
                       FStr = object(FText)
                       FNum = object(FText)
                         FReal = object(FNum)
                         FInt = object(FNum)
                           FZip = object(FInt)
Sliders          FSlider = object(Field)
Odemo          TestNode = object(Node)
                   IntNode = object(TestNode)
                   StrNode = object(TestNode)
TCCell       Cell = object(Base)
                 EmptyCell = object(Cell)
                 ValueCell = object(Cell)
                 TextCell = object(Cell)
                 FormulaCell = object(Cell)
                 RepeatCell = object(Cell)
```

U levom stupcu je ime datoteke npr. **Objects**, a u desnom stupcu je ime objekta i oznaka (eventualne) roditeljske klase. **SProc = object** znači da klasa **SProc** nema roditelja, dok **MenuItemProc = object(MenuItem)** znači da klasa **MenuItemProc** nasleđuje klasu **MenuItem**. Nasleđivanje se označava uvlačenjem reda. Događa se da jednu istu klasu nasleđuju klase iz različitih jedinica, kao i da se više klasa nalazi u istoj jedinici. Na primer, definicija klase **Location** je u jedinici **Objects**, klasa **Cursor** je u istoj jedinici a nasleđuje **Location**, dok je **Block** u jedinici **Bricks** ali takođe nasleđuje **Location**.

Bilo bi vrlo pogodno da postoji program koji bi pravio gornju hijerarhiju klasa. Tako bi uvek znali šta se nasleđuje i od koga. Ulazne podatke u taj program bi mogao napraviti GREP.

Za *Turbo Pascal* 5.5 je karakteristično da nema jednu klasu iz koje bi sve druge izvirale. Ako nam treba, u svakom trenutku je možemo sami napraviti ponovnim prevođenjem.

## OBJEKAT Base

Objekat **Base** je apstraktan tip podataka i služi samo kao zajednička osnova za druge objekte. To istovremeno znači i da ne treba da postoji nijedan primerak ove klase. Naslednicima klase **Base** garantuje se samo postojanje destruktora **Done**.

U jedinici **Objects** nalaze se dve osnovne objekta: **Stream** i **List**. Objekat **Stream** odgovara pojmu datoteke u običnom *Turbo Pascal*-u. Tip **file of object** nije dozvoljen pa je **Stream** prirodna dopuna. Objekat **List** je jednostruko povezana lista objekata. Svaki element te liste mora biti tipa **Node** ili izveden iz tipa **Node**.

## OBJEKAT Stream

Slično netipiziranim datotekama, **Stream** podržava osnovne operacije sa datotekama (otvaranje, zatvaranje, čitanje, upis i pretraživanje). Za razliku od netipizirane datoteke, **Stream** može da polimorfno čita i piše objekte. Za to je potrebno da novi objekat definiše metode **Store** i **Load** i da se registruje kao korisnik tipa **Stream**.

**Stream** je osnovna klasa za sve druge vrste tokova. Najveći deo metoda je apstraktan i postoji samo da bi naslednićke klase mogle da ih promene.

Klasa **Stream** ima sledeće promenljive:

**TypeCount** – broj registrovanih tokova

**TypeList** – pokazivač na niz adresa registrovanih tipova u VMT tabeli

**ProcList** – pokazivač na niz **Store** i **Load** pokazivača na metode

**Status** – status toka. Vrednost **Status**<>0 znači da se dogodila greška u radu sa tokom i sve potonje ulazno/izlazne operacije se ignorišu. Promenljiva **Status** ima istu ulogu kao i standardna funkcija **IOResult**, osim što se **Status** mora „ručno" menjati da bi se omogućile daljnje ulazno-izlazne operacije.

Konstruktori i destruktori klase **Stream** su

```
constructor Init;
```

Zauzima memoriju za **TypeList** i **ProcList**, zove **RegisterTypes** da bi registrovao tipove koji su vezani na tok.

```
destructor Done; virtual;
```

Uklanja **TypeList** i **ProcList**.

Osnovni ulazno-izlazni metodi klase **Stream** su

```
procedure Read(var Buf; Count: Word); virtual;
```

Čita **Count** bytove iz toka u **Buf**. U slučaju greške **Buf** se puni nulama. Ovaj metod mora biti izmenjen u naslednićkoj klasi.

```
procedure Write(var Buf; Count: Word); virtual;
```

Upisuje **Count** bajtova iz **Buf** u tok. Ovaj metod mora biti izmenjen u naslednićkoj klasi.

```
procedure Flush; virtual;
```

Snima podatke iz memorije na disk. Ovaj metod mora biti izmenjen.

```
procedure Truncate; virtual;
```

Prekida tok na tekućoj poziciji, tj. tekući položaj pokazivača na tok postaje kraj toka. Ovaj metod mora biti izmenjen.

```
function GetPos: Longint; virtual;
```

Vraća tekući položaj toka, ili -1 u slučaju greške.

```
procedure SetPos(Pos: Longint; Mode: Byte);
virtual;
```

Postavlja pokazivač toka na novi položaj koji je za **Pos** bajtova udaljen od položaja u **Mode**. Moguće vrednosti u **Mode** se dobijaju iz konstanti **PosAbs**, **PosCur** i **PosEnd** koje predstavljaju početak, tekuću poziciju i kraj toka. Ovaj metod mora biti izmenjen.

```
function GetSize: Longint;
```

Vraća trenutnu dužinu toka. Poziva **SetPos** i **GetPos**. Ovaj metod ne sme biti izmenjen.

```
procedure Seek(Pos: Longint);
```

Traži zadatu poziciju. To je zapravo poziv metoda **SetPos** sa vrednošću **PosAbs** u **Mode**. Ovaj metod ne sme biti izmenjen.

Metodi za registrovanje tipova u klasi **Stream** su

```
procedure RegisterTypes; virtual;
```

Registruje sve tipove koje **Stream** mora da zna. Registrovanje objekta kod toka se obavlja nadjačavanjem ovog metoda za svaki novi tip. U okviru nasleđenog metoda **Register** prvo treba zadavati nasleđene tipove u **RegisterTypes** da bi se registrovali tipovi koji su registrovani kod predaka. Tip treba registrovati samo ako će njegovi primerci biti čitani i pisani metodima **Get** i **Put**. Ako nije izmenjen, ovaj metod ne radi ništa.

```
procedure Register(TypePtr, StorePtr, LoadPtr: Pointer);
```

Registruje tip kod toka. Ovaj metod se sme upotrebiti samo u metodu **RegisterTypes**. Poziva se ovako

```
Register(TypeOf(AType), @AType.Store,
@AType.Load);
```

gde je **AType** tipa izveden iz objekta **Base**, tj. **AType** za svog pretka ima **Base**. **AType** mora da definiše **Store** i **Load** metode sa sledećim zaglavljima

```
procedure Store(var S: Stream);
constructor Load(var S: Stream);
```

Metod **Store** mora pisati binarni oblik objekta na tok S (pomoću **S.Write**); konstruktor **Load** čita taj binarni oblik sa toka, koristeći **S.Read**.

Polimorfni ulazno-izlazni metodi klase **Stream** su

```
procedure Put(P: BasePtr);
```

Upisuje dati objekat u tok. Objekat se prethodno mora registrovati (izmenjeni metod **RegisterTypes**).

```
function Get: BasePtr;
```

Čita objekat sa toka i vraća pokazivač na njega. Može da čita samo objekte koji su upisani sa **Put**.

Klasa **Stream** prijavljuje greške procedurom **Error**

```
procedure Error(Code: Integer); virtual;
```

Ovaj metod se poziva kadgod se pojavi greška u radu sa tokom. **Code** sadrži 16-bitni kod greške. Ovaj metod u svom osnovnom obliku samo postavlja **Code** u varijablu **Status**.

## OBJEKAT DosStream

DosStream je tok koji radi sa DOS-om, tj. piše i čita DOS datoteke. Njegova jedina promenljiva je

```
Handle: Interni broj datoteke u DOS-u
```

Konstruktori i destruktori klase **DosStream** su.

```
constructor Init(FileName: FNameStr; Mode:
Word);
```

Poziva **Stream.Init** a zatim otvara datoteku sa imenom **FileName** i atributom pristupa iz **Mode**. Prihvatljive vrednosti za **Mode** su konstante **SCreate**, **SOpenRead**, **SOpenWrite**, **SOpen**. **SCreate** stvara novu datoteku. Ostale tri operacije otvaraju datoteku samo za čitanje, samo za upis ili i za jedno i za drugo. Broj datoteke pod DOS-om je u promenljivoj **Handle**.

```
destructor Done; virtual;
```

Zatvara datoteku i poziva **Stream.Done**.

Osnovni ulazno/izlazni metodi klase **DosStream** su:

```
procedure Read(var Buf; Count: Word); virtual;
```

Metod **Read** za DOS datoteke. Učitava **Count** bajtove u **Buf** pomoću DOS funkcije 3FH. U slučaju greške, puni **Buf** nulama.

```
procedure Write(var Buf; Count: Word); virtual;
```

Metod **Write** za DOS datoteke. Pozivom DOS funkcije 40H – počev od memorijske adrese **Buf** – upisuje **Count** bajtove u datoteku.

```
[illegible]cedure Truncate [illegible]
```

Pozivom DOS funkcije 40H odseca DOS datoteku

```
function GetPos  Longint  virtual
```

Metod **GetPos** za DOS datoteku. Poziva DOS funkciju 42H

```
procedure SetPos(Pos  Longint  Mode  Byte)
virtual
```

Metod **SetPos** za DOS datoteku  Poziva DOS funkciju 42H

```
procedure Open(var Name  Mode  Word)
```

Privatni metod koji stvara ili otvara datoteku pozivom DOS funkcije 3CH ili 3DH  Poziva se iz **DosStream Init** ovu proceduru nikada ne treba izvršavati direktno

```
procedure Close;
```

Privatni metod koji snima sve delove datoteke koji postoje eventualno nesnimljeni u memoriji  Poziva **Flush** i zatvara datoteku. Poziva se iz **DosStream Done**; ovu proceduru nikade ne treba zvati direktno

## PRIMER ZA DosStream

Klasa **DosStream** se ne može upotrebljavati samostalno Sledeći program pokazuje sve što možemo napisati a da ne definišemo objekat koji radi sa diskom preko klase **DosStream**

```
program [illegible]
[illegible]
var
  [illegible]
begin
    tok init('imedat txt   SCreate)
    [      naredbe za upis i ispis podataka
    ]
    tok done
end
```

Umesto klase **DosStream** skoro uvek ćemo koristiti klasu **BufStream.**

## KLASA BufStream

Klasa **BufStream** je slična klasi **DosStream** osim što se podaci iz memorije prenose na disk u blokovima čiju dužinu zadaje sam programer. (Blok za prenos zove se bafer.) Primena ove klase znatno ubrzava programe koji sadrže veliki broj naredbi **Read** i **Write** sa malom dužinom podataka. Preporučuje se dužina od 1024 bajta.

Promenljive klase **BufStream** su

**Buffer** Pokazivač na bafer
**BufSize** Veličina bafera
**BufPtr** Adresa bajta pročitanog iz bafera.
**BufEnd** Pokazivač na tekući bajt u baferu  Ako je **BufPtr=BufEnd** bafer je prazan. Ako je **BufPtr<BufEnd,** bafer je u režimu čitanja i bajtovi između **BufPtr** i **BufEnd** su pročitani unapred iz datoteke Kada je **BufPtr>BufEnd**, datoteka je u režimu upisa i bajtovi između **BufEnd** i **BufPtr** su upisani u datoteku ali još nisu snimljeni na disk

Konstruktori i destruktori klase **BufStream** su:

Rezerviše bafer dužine **Size** i poziva **DosStream Init.**

Poziva **DosStream.Done** i uklanja bafer
Osnovni ulazno izlazni metodi toke su

Metod **Read** za ovu klasu. Bafer je dugačak **BufSize** bajtova.

Metoda za **Write** za ovu klasu. Bafer je dugačak **BufSize** bajtova.

Snime sve podatke na disk.
Evo primera za upotrebu klase **BufStream.**

```
uses Objects, Crt [illegible]
type
  adresaTeksta [illegible]
  Txt = object
    ime   string
    txtPok  adres[illegible]
    constructor Init ([illegible]
    destructor D[illegible]
    [illegible]
    procedure [illegible]
    procedure St[illegible]
  [illegible]
```

```
  Stream R
  Register(Typ

begin
  ClrScr
  newTxt init(
  newTxt wr
  oldTxt init(    oldTxt')
  S Init( NEWSTR DTA'  SCrea
  newTxt store(S)
  S done
  S Init( NEWSTR DTA , SOpen
  oldTxt load(S)
  oldTxt wr
  wait  = getKey;
  with S  newTxt, oldTxt do done
end
```

Program DosDemo definiše klasu Txt koja nije ništa drugo do standardni tip string prepisan kao objekat sa najnužnijim naredbama za rad sa tokovima. Txt služi samo za ilustraciju tokova a u praksi bi smo normalno ovde objašnjene naredbe load i store napisali direktno u klasi VelikiTekst ili nekoj sličnoj.

Promenljiva klase Txt su ime koja sadrži ime promenljive tipa Txt radi lakše identifikacije u metodu wr. txtPok je pokazivač na tekst. Konstruktor init zauzima memorijski prostor za tekst pomoću getMem i postavlja obe promenljive. Destruktor done oslobađa zauzetu memoriju pozivom procedure freeMem. Da bi objekat Txt mogao da koristi usluge nekog toka, moramo

1) napisati metode load i store koji binarno pišu i čitaju sa diska,

2) definisati tok tipa BufStream i u njemu

3) napisati samo jednu proceduru, po imenu RegisterTypes. Oblik te procedure je isti iz programa u program i ne treba ga menjati. Preciznije, možemo ga menjati ali onda se i imena metoda za binarno čitanje i pisanje moraju promeniti. Ako želimo da umesto imena metoda load i store bude npr. upis i citanje, onda drugi red u metodu Tok.RegisterTypes treba da glasi:

```
Register(TypeOf(Txt), @Txt upis  @Txt cita
nje)
```

Uobičajeno je da se ti metodi zovu load i store, i te konvencije se pridržavamo u ovom tekstu.

U metodu store moramo kao naredbu za snimanje na disk upotrebiti metod write, a u metodu load moramo upotrebiti metod read. Metodi write i read odnose se na definisani tok (u gornjem programu tok je u promenljivoj S) i nasleđuju se iz roditeljskih klasa – ne pišemo ih sami.

Iz ovog primera može se (pogrešno) zaključiti da za svaku klasu treba da uvedemo poseban tok radi komunikacije sa diskom. To nije tačno: isti tok može da opsluži proizvoljan broj objekata, ali ih prethodno moramo registrovati kod tog toka. To registrovanje se svodi na prosto dodavanje programskih redova u metod Tok.RegisterTypes. Na primer, neki objekat noviObjekat bi registrovali dodavanjem naredbe

```
Register(TypeOf(noviObjekat) @novi
Objekat Store  @noviObjekat Load)
```

pod pretpostavkom da se metodi za binarno obraćanje disku i u njemu nazivaju store i load.

Datoteku na disku kreira konstruktor klase Tok: S.Init. Prvi parametar konstruktora je ime DOS datoteke, ovde je to NEWSTR.DTA. Drugi parametar, SCreate, je jedan od četiri moguća parametra na tom mestu. SCreate samo stvara datoteku i mora biti upotrebljen u prvom konstruktoru u programu. Međutim, ako želimo da kasnije nešto radimo sa tim tokom (tj. sa datotekom) – moramo prvo ukinuti inkarnaciju tog toka, tj. primeniti destruktor S.done. Zatim *ponovo* otvaramo isti tok u istoj (ili nekoj drugoj) promenljivoj ali sada sa nekim od preostala tri parametra – samo za upis, samo za čitanje ili za oba. U našem primeru drugi parametar u drugom konstruktoru je SOpen, što nam dozvoljava da sa oldTxt load(S) očitamo sadržaj toka u promenljivu oldTxt.

Ograničenje rada sa tokovima proističu iz obaveze da metodi load i store čitaju i pišu binarno. Datoteka koja na disku predstavlja tok ne može biti duža od 64K zbog segmentne organizacije Intelovih procesora iz porodice 80x86. Doduše, na procesorima tipa 80386 segment može biti dugačak nekoliko megabajta ali *Turbo Pascal 5.5* to ne podržava eksplicitno. Ništa nas ne sprečava da grupe podataka duže od 64K na disku predstavimo kao niz datoteka dužine do 64K i da na taj način doskočimo ograničenju PC procesora.

## TIP Node

Node je apstraktni tip za elemente u povezanim listama tipa List. Polje Next pokazuje na sledeći element lista. Ako je element poslednji, polje Next pokazuje na početak liste, tako da je lista kružna. Metod Prev vraća pokazivač na prethodni element. Za prvi element liste, Prev pokazuje na poslednji element u listi.

## TIP List

Objekat List pruža osnovne algoritme za povezanu kružnu listu. Elementi liste mora biti apstraktnog tipa Node ili iz nekog tipa koji nasleđuje Node.

Objekat List ne nasleđuje ništa tako da nili ime virtuelnih metoda nili ime konstruktora i destruktora.

Promenljive u klasi List je:

Last  Pokazivač na poslednji element liste, odnosno ima vrednost NIL ako je lista prazna. Izraz Last^ Next pokazuje na prvi element liste.

Metodi tipa List su:

```
procedure Clear
```

Briše sadržaj liste postavljanjem polja Last na NIL. Eventualno postojeći elementi se ne brišu fizički.

```
procedure Delete
```

Uklanja sve elemente liste primenjujući na svakog destruktor Done.

```
procedure Append(N  NodePtr)
```

Dodaje element listi. Novi element postaje prvi u listi.

Umeće element. Novi element postaje poslednji u listi.

```
procedure Remove(N: NodePtr);
```

Briše element liste. Element se ne uklanja fizički.

```
function First: NodePtr
```

Vraća pokazivač na prvi element liste ili vrednost nil ako je lista prazna. Poslednji element liste se može dosegnuti kroz polje **last**.

```
function Next(N: NodePtr): NodePtr;
```

Vraća pokazivač na element posle N ili NIL ako je N poslednji element. Isto što i N^.next, osim što polje next poslednjeg elementa pokazuje na prvi element u listi.

```
function Prev(N: NodePtr): NodePtr;
```

Vraća pokazivač na element pre N, ili NIL ako je N prvi element. Isto što i N^.prev, osim što metod prev za prvi element vraća poslednji element u listi.

```
function Empty: boolean;
```

Vraća **true** ako je lista prazna, inače vraća **false**.
Ulazno-izlazni metodi klase **List** su:

```
procedure Store(var S: TStream);
```

Postavlja listu u tok. Svi tipovi elemenata koje lista sadrži prethodno moraju biti registrovani za taj tok. Lista se ubacuje u tok tako što se šalje poruka **S.put** svakom elementu liste, a iza toga sledi **S.put(NIL)**.

```
procedure Load(var S: TStream);
```

Učitava listu iz toka. Svi tipovi elemenata koje lista sadrži prethodno moraju biti registrovani za taj tok. Lista se učitava dodavanjem rezultata poruke **S.get** sve dok **S.get** ne vrati vrednost NIL.

## PRIMER ZA SPREGNUTE LISTE

```
program LDemo
  [illegible]
  ...
  obj...
  ObjLonjint
  [illegible]
    P := ...
    P := ...ojB...Next...;
  end
end;
begin
  C...
  writeln(...
  sto... lca;
  ista.Append(New(...
  lista.Append(New(...
  lista.Append(New(...
  lista.Append(New(CeoBroj...
  ...wr...
  ...
end.
```

Klasa **CeoBroj** nasleđuje **Node** i kao takva može biti element liste. Klasa **TestList** nasleđuje **List** i samo dodaje metod **wr** za ispisivanje sadržaja liste. Taj metod pokazuje kako se dolazi do nekog čvora liste: za razliku od niza, u listi nema indeksiranja. Pokazivač na objekat tipa **CeoBroj**, tip **CeoBrojPok**, ima glavnu ulogu u **while** petlji.

**lDemo** je najjednostavniji primer liste jer je zastupljen samo jedan tip, **CeoBroj**, kao čvor liste. Prava vrednost lista i polimorfizma vidi se tek definisanjem većeg broja tipova kao čvorova liste. Videti Borlandov program ODEMO.QAS sa jedne od disketa, u kome su celi brojevi i tekst ravnopravni čvorovi liste.

## OBJEKTI Forms i Field

U jedinici **Forms** nalaze se dva objekta, **Field** i **Form**. **Field** je polje za unos podatka a **Form** je niz takvih polja koja zajedno čine ulaznu masku.

Jedinica **Forms** sadrži hijerarhiju različitih polja za unos podataka. Na vrhu je apstraktna klasa **Field**, zatim slede **FText** (apstraktno tekstualno polje), **FStr** (polje za unos teksta), **FNum** (apstraktno polje za unos numeričkih podataka), **FInt** (podaci tipa **integer**), **FZip** (poštanski broj), **FReal** (broj tipa real). Hijerarhija izgleda ovako:

```
Field
  ──────→ FText
              ──────→ FStr
              ──────→ FNum
                          ──────→FInt
                                    ──────→FZip
                          ──────→FReal
```

## KLASA Field

Iako apstraktna klasa **Field** je potomak klase **Node**. Promenljiva tipe **Field** su

- **X** X koordinata polje
- **Y** Y koordinata polje
- **Size** Veličina podatka smeštenog u polju,
- **Title** Pokazivač na tekst naslova
- **Value** Pokazivač na trenutnu vrednost (dugačku **Size** bajtova)
- **Extra** Označava položaj polja definisan u naslednićkim tipovima

Metodi objekta **Field** su

- **Init** Stvara i priprema polje
- **Load** Učitava polje iz toka,
- **Done** Briše i uklanja polje iz memorije
- **Clear** Postavlja polje na osnovnu vrednost;
- **Edit** Unos i izmena polja, vraća kraj reda,
- **Show** Prikazuje polje
- **Store** Upisuje polje u tok

## POTKLASE ZA Field

Klase **FText** je apstraktna klasa za sve polje kojime se učitava tekst. Ona dodaje metode **GetStr** i **PutStr** konkretizuje metode **Edit** i **Show**. **GetStr** pretvara sadržaj polja u tekst, a **PutStr** je inverzna operacija za **GetStr**. **PutStr** vraća **True** ako je u tekstu dozvoljena vrednost. U protivnom vraća **False** i ne menja vrednost polja.

- **FStr** je polje tipa **String**, tj. tekst promenljive dužine
- **FNum** je potklasa apstraktne klase ali je i sama za sebe apstraktna za sve numeričke ulazne podatke. U njoj se menja metod **Show** i to tako da poravnava vrednost polje udesno
- **FInt** učitava tip **Longint** pri čemu programer zadaje gornju i donju vrednost podatka
- **FZip** je poštanski broj, a to nije ništa drugo nego polje tipa **FInt** se samo 5 cifara.
- **FReal** je polje za broj sa pokretnim zarezom, programer sam definiše njegovu ukupnu dužinu i broj decimala.

## TIP Form

Objekat **Form** je ulazna maska. Realizovan je kao lista pri čemu su čvorovi liste polje različitih tipova. Promenljive tipe **Form** su

- **X1** Leva koordinata prozora,
- **Y1** Gornja koordinata prozora;
- **X2** Desna koordinata prozora
- **Y2** Donje koordinata prozora
- **Size** Ukupna dužina podata u maski – zbir promenljivig **Size** za sve maske u listi
- **Fields** Lista objekta **Field**

Metodi su

- **Init** Stvara i priprema masku za rad
- **Load** Učitava masku iz toka
- **Done** Briše i uklanje masku
- **Edit** Unos i izmena maske i vraća znak za kraj unosa,
- **Show** Prikazuje masku, eventualno prvo briše ceo prozor
- **Add** Dodaje polje u masku
- **Clear** Postavlje celu masku na osnovnu vrednost
- **Get** Kopira masku u promenljivu
- **Put** Kopira promenljivu u masku
- **Store** Upisuje masku u tok.

Metodi **Get** i **Put** kopiraju podatke u masku i iz maske. Promenljive koje se prenosi ovim procedurama mora u svemu odgovarati polj me iz maske

Tip **FStream** nasleđuje **BufStream** da bi mogao da radi sa tipovima **FStr**, **FInt**, **FZip** i **FReal**

Osim ovih objekata, po različitim demonstracionim programima nalaze se i mnogi drugi tipovi: **Location**, **Point**, **Circle**, **Block**, **HashTable**, **Cell**, **Obstacle**, **Wall**, **Arc**, **InputField**, **ProgramObject**, **ParserObj**, **Menu**, **MenuItem**, **CardList**, **TextString**, **Counter**, **SpreadSheet**, **Screen**, **ScreenArea**, **ColorTable**, **Cursor**, **SaveScreen**, **Block**, **Ball**, **Brick** itd

## PRIMER ZA ULAZNE MASKE

```
program MaskaB,
uses Crt, Forms;
type
  Osoba = record
    ime             string[30];
    prezime         string[30],
    ulica           string[32],
    grad            string[16]
    postanskiBroj   Longint,
    end,
const
  podaci: Osoba = (
    ime             : 'IME'
    prezime         : 'PREZIME'
    ulica           : 'ULICA';
    grad            : 'GRAD',
    postanskiBroj : 111);
var
  maska  Form;
  neko : Osoba,
begin
  ClrScr,
  writeln('Program MASKA');
  maska Init(10  10, 56, 17);
  maska Add(New(FStrPtr, Init(3, 2, ' ime
', 30)));
  maska Add(New(FStrPtr, Init(3, 3, ' prezi-
me  ', 30)));
```

```
  maska Add(New(FStrPtr, Init(3, 5, ' ulica
', 32)));
  maska Add(New(FStrPtr, Init(3, 6, ' grad
', 16)));
  maska Add(New(FZipPtr, Init(3, 7, ' p
broj  ')));
  neko := podaci;
  maska Put(neko);
  maska Show(True);
  if maska Edit = CSave then maska.Get(neko);
  maska Done;
end. { MaskaD }
```

Program **MaskaD** definiše tip podataka **Osoba** sa osnovnim podacima o nekoj osobi. Konstantu **podaci** postavljamo na pomalo neubičajen ali vrlo konstan način - direktnim pridodeljivanjem konstanti delovima sloga. Zato u programu vrednost promenljive **neko** tipa **Osoba** postavljamo elegantno sa **neko:=podaci**. Promenljiva **maska** je tipa Form iz klase **Forms**. Njen **init** metod zadaje gornje levo i donje desno teme pravougaonika u kojem će se maska pojaviti. Taj pravougaonik je istaknut posebnom bojom. Metod **Add** dodaje polja u masku, jer **maska** nije ništa drugo do (spregnuta) lista polja. Argument za **Add** je kompleksan: prvo se izvršava naredba **init** za polje npr. tipa **FStrPtr** zatim se dodeljuje memorijski prostor pokazivaču kojeg vraća naredba **new**. U ovom programu se koristi proširenje naredba **new** inače bi trebalo pisati u jednom redu metod **init** a u drugom redu naredbu **new**. Promenljiva **maska** je takođe primer liste u kojoj su čvorovi različitih tipova (prvo četiri puta tekst a poslednje polje je dugačak ceo broj).

Sadržaj postavljamo u masku metodom **put** a prikazujemo je metodom **show**. Naredba **if** proverava da li postoji neka izmena u izloženim podacima pa ako je ima - učitava je u sadržaj maske (metod **get**). Na kraju programa destruktorom **done** uništavamo masku.

Maske tipa **Forms** su primer kako se jednostavno koriste „tuđe" klase tj. klase koje je neko kvalitetno napisao i provario. Osim postavljanja podataka na ekran korisnik klase **Forme** ne mora ništa da zna o načinima kretanja kursora, o ispitivanju tastature na PC računarima ili o bojama na ekranu. Tip **Forma** je po svemu ekvivalentan ugrađenim tipovima *Turbo Pascal* a.

## APSTRAKTNA KLASA Velicina

Klasa **Velicina** je apstraktna klasa za sve objekte koji se mogu porediti i meriti, odnosno, nad kojima osnovne aritmetičke operacije imaju smisla. Njene potklase su obični brojevi ali i vreme, datum i drugi. Ovde su kao potklase iznova napisani standardni tipovi **char**, **real** i **integer** u obliku klasa **Znak**, **Float** i **Int**. Klasa **Broj** nasleđuje klasu **Velicine** ali samo kao još jedne apstraktne klase koju nasleđuju pravi brojevi. Objekat **Broj** obogaćuje klasu **Velicine** elementarnim funkcijama kao što je dizanje na stepen, kvadratni koren i druge. Konačno, **Vreme** je klasa koja po svemu zaslužuje poseban tretman, ali ga u klasičnim programskim jezicima nikada name. (Slično bi se moglo reći i za klasu **Datum** koja ovde nije data.)

Jedinica **DataLib0** služi kao zajedničko mesto za sve konstante koje se mogu primenjivati u drugim modulima. Evo je:

```
unit datalib0
interface
const max_string = 255
      line_long = 80
type string255 = string[max_string]
     string128 = string[128]
     string98 = string[98]
     string80 = string[80]
     string64 = string[64]
     string40 = string[40]
     string20 = string[20]
     string10 = string[10]
     complex = record
             treal, timag   real
             end
     tipPodatka = integer;
implementation
end  { Datalib0 }
```

Objekat **Velicina** sadrži standardne metode **init**, **done**, **wr**, **wrln** a konceptualno novi su **manjeod**, **ednako**, **između**, **max** i **min**. Nema nikakvog razloga (osim nedostatka prostora) da se ovaj protokol ne dopuni i drugim operacijama poređenja.

**Znak**, **Vreme** i **Broj** nasleđuju direktno klasu **Velicine**. **Znak** je samo preodevani tip **char**. Unutar njega poredi se preko standardne funkcije **ord**. **Vreme** je realizovano preko pokazivača na slog koji sadrži sate i minute. Poredi se konverzijom ta dva broja u minute tokom dana. Učitava se iz dva puta, prvo sat pa minuti.

**Broj** dodaje samo jednu aritmetičku operaciju **plus** i jednu elementarnu funkciju **exponent**. Svakako da postoji na desetine numeričkih operacija koje bi ovoj klasi i njenim naslednicima trebalo dodati. Objekat **Float** osim uobičajenih metoda za poređenje i numeriku dodaje i dva za učitavanje, **read** i **read_limit**. Prvi učitava tekst i pretvara ga u broj ako je ispravan, a drugi završava učitavanje samo ako je broj između dva u parametara. Potpuno je ista situacija i sa klasom **Int** koja predstavlja cele brojeve. Konačno, glavni program prikazuje mogućnosti većine izloženih metoda.

```
program vel1;
uses crt, Datalib0
type
{  ----------- objekat Velicina ---------- }
    Velicina = object
      procedure init;
      procedure done
      function manjeod   boolean;
      function jednako : boolean
      function izmedju   boolean;
      function max : boolean;
      function min : boolean;
      procedure wr;
      procedure wrln;
      end; { velicina }
      procedure velicina init     begin end;
      procedure velicina done;    begin end;
      function velicina manjeod   begin end;
      function velicina jednako;  begin end
      function velicina izmedju,  begin end
      function velicina max       begin end;
      function velicina min       begin end
      procedure velicina wr;      begin end
      procedure velicina wrln     begin end,
{ ---------- objekat Znak    ------------- }
type
    Znak = object(Velicina)
      c : char,
      constructor init
      destructor done; virtual;
      function manjeod (ch : char) : boolean;
virtual,
      function jednako (ch : char) : boolean
virtual;
      function izmedju (c1  c2 : char) : boole-
an, virtual
      function max (ch : char)   char      vir
tual
      function min (ch : char)   char
virtual;
      procedure postaje(ch   char)
      function je : char;
```

```
      procedure wr;
      end; { Znak }
constructor Znak.init
begin
    c := chr(0);
end
destructor Znak.done;
begin end
function Znak.manjeod;
begin
    manjeod := ord(c) < ord(ch)
end
function Znak.jednako;
begin
    jednako := ord(c) = ord(ch)
end;
function Znak.izme'u
begin
    izme"u := (ord(c) > ord(c1)) and ((ord(c)
< ord(c2));
end
function Znak.max
begin
    if ord(c) <= ord(ch)
       then max := ch
       else max := c
end
function Znak.min;
begin
    if ord(c) <= ord(ch)
       then min := c
       else min := ch
end;
procedure Znak.postaje
begin
    c := ch;
end;
function Znak.je;
begin
    je := c
end;
procedure Znak.wr;
begin
    write(c);
end
{       -------  objekat Broj  ---   ---- }
type
   Broj = object(Velicina);
     function exponent : real;
     function plus : word
     end; { Broj }
function Broj.exponent
begin
end
function Broj.plus;
begin
end;
{ -------        objekat Float        ----- }
type
   Float = object(Broj);
     f : real;
     function exponent : real ; virtual;
     constructor init;
     destructor done  virtual;
     function manjeod (f1 : real) : boolean;
virtual,
     function jednako (f1 : real) boolean;
virtual
     function izme'u (f1,f2 : real); boolean;
virtual
     function max(f1:real) : real  virtual;
     function min(f1:real) : real  virtual
     procedure postaje(f1 : real)
     function je : real; virtual,
     function plus(f1 : real) : real; virtual
     procedure read; virtual;
     procedure read_limit (low, high : real
); virtual;
     procedure wr; virtual;
     procedure wrln; virtual,
     end; { Float }
function Float.exponent
begin
    exponent:= exp(f);
end;
constructor Float.init;
begin
    f := 0.0;
end;
destructor Float.done
begin
end;
function Float.manjeod;
begin
    manjeod := f < f1;
end;
function Float.jednako
begin
    jednako := f = f1
end
function Float.izme"u,
begin
    izme"u := (f > f1) and (f < f2);
end;
function Float.max,
begin
    if f <= f1
       then max := f1
       else max := f
end
function Float.min;
begin
    if f <= f1
       then min := f
       else min := f1
end
procedure Float.postaje
begin
    f := f1
end;
function Float.je
begin
    je := f
end
function Float.plus;
begin
    plus := f + f1;
end;
procedure Float.read;
const beep = ^g
var error_code, xpos, ypos : integer;
    strng : string80;
begin
    xpos := wherex
    ypos := wherey
    readln(strng);
    val(strng, f, error_code)
    while error_code > 0 do begin
          write(beep);
          gotoxy(xpos, ypos); clreol;
          readln(strng)
          val(strng, f, error_code)
    end
end; { Float.read_int }
procedure Float.read_limit(low, high : real )
const beep = ^g;
var error_code, xpos, ypos : integer
    temp : real;
    strng : string80;
begin
    if low > high then begin
       temp := low
```

```
        low := high;
        high := temp
      end;
      xpos := wherex;
      ypos := wherey;
      readln(strng);
      val(strng, f, error_code);
      while (error_code > 0) or
            (f < low) or (f > high) do begin
            write(beep);
            gotoxy(xpos, ypos); clreol;
            readln(strng);
            val(strng, f, error_code)
      end;
end; { Float read_limit }
procedure Float wr;
begin
     write(f)
end;
procedure Float wrln;
begin
     writeln(f);
end;
{ ----- ----- objekat Int ----- ------- }
type
    Int = object(Broj)
      i : integer;
      function exponent : real; virtual;
      constructor init;
      destructor done;
      function manjeod (i1 : integer) : boole-
an
      function jednako (i1 : integer) : boole-
an
      function izme'u (i1 i2 : integer) : boo-
lean;
      function max (i1 : integer) : integer;
      function min (i1 : integer) : integer;
      procedure postaje(i1 : integer);
      function je : integer
      function plus(i1 : integer) : integer;
      procedure read
      procedure read_limit(low, high : integer
);
      procedure wr
      procedure wrln
      end { Int }
function Int exponent;
begin
     exponent := exp(i);
end
constructor Int init
begin
     i := 0;
end;
destructor Int.done;
begin
end
function Int manjeod;
begin
     manjeod := i < i1
end;
function Int.jednako;
begin
     jednako := i = i1
end;
function Int izme"u;
begin
     izme u := (i>i1) and (i<i2)
end;
function Int.max;
begin
     if i <= i1
        then max := i1
        else max := i
end;
function Int min;
begin
     if i <= i1
        then min := i
        else min := i1
end;
procedure Int postaje;
begin
     i := i1
end;
function Int je;
begin
     je := i
end;
function Int plus;
begin
     plus := i + i1;
end
procedure Int read;
const beep = ^g;
var error_code, xpos  ypos : integer;
    strng : string80;
begin
     xpos := wherex;
     ypos := wherey;
     readln(strng);
     val(strng, i, error_code);
     while error_code > 0 do begin
           write(beep);
           gotoxy(xpos, ypos)  clreol
           readln(strng);
           val(strng, i, error_code);
     end
end { Int read }
procedure Int read_limit(low, high : integer);
const beep = ^g
var error_code, temp, xpos, ypos : integer;
    strng : string80;
begin
     if low > high then begin
        temp := low;
        low := high
        high := temp
     end
     xpos := wherex;
     ypos := wherey
     readln(strng);
     val(strng, i  error_code)
     while (error_code > 0) or
           (i < low) or (i > high) do begin
           write(beep);
           gotoxy(xpos, ypos); clreol;
           readln(strng);
           val(strng, i, error_code)
     end;
end { Int read_limit }
procedure Int wr;
begin
     write(i);
end
procedure Int wrln
begin
     writeln(i);
end;
{ ----------- objekat Vreme --------- }
type
      vrPok = ^vr;
      vr = record
         sat  minut : integer;
         end;
      Vreme = object(Velicina)
        v : vrPok;
        constructor init;
        destructor done;
        function manjeod(v1  vrPok) : boolean;
        function jednako(v1  vrPok) : boolean;
```

```
        function izme"u(v1, v2 : vrPok) : boo-
lean;
        function max(v1 : vrPok) : vrPok;
        function min(v1 : vrPok) : vrPok;
        procedure postaje(s, m : integer);
        function je : vrPok;
        function plus(v1 : vrPok) : vrPok;
        function u_minute(v1 : vrPok) : inte-
ger
        function iz_minute(minuti : integer) :
vrPok;
        procedure wr; virtual;
        procedure wrln; virtual;
        procedure read; virtual;
        end; { Vreme }
constructor Vreme.init;
begin
    new(v);
end;
destructor Vreme.done;
begin
    dispose(v);
end;
function Vreme.manjeod;
begin
    manjeod := u_minute(v) < u_minute(v1);
end;
function Vreme.jednako;
begin
    jednako := u_minute(v) = u_minute(v1);
end;
function Vreme.izme"u;
begin
    izme"u := (u_minute(v) > u_minute(v1)) and
              (u_minute(v) < u_minute(v2));
end;
function Vreme.max;
begin
    if u_minute(v) <= u_minute(v1)
        then max := v1
        else max := v
end;
function Vreme.min;
begin
    if u_minute(v) <= u_minute(v1)
        then min := v
        else min := v1
end;
procedure Vreme.postaje;
begin
    v^.sat := s;
    v^.minut := m;
end;
function Vreme.je;
begin
    je := v
end;
function Vreme.u_minute;
begin
    u_minute := v1^.sat * 60 + v1^.minut;
end;
function Vreme.iz_minute;
begin
    v^.sat := minuti mod 60;
    v^.minut := abs(minuti - v^.sat) mod 60;
    iz_minute := v
end;
function Vreme.plus;
var s, m : integer;
    vp : vrPok;
begin
    s := v^.sat + v1^.sat;
    m := v^.minut + v1^.minut;
    if m > 60 then begin
        m := m - 60;
        s := s + 1;
        end; { if }
    if s > 24 then s := s - 24;
    new(vp);
    vp^.sat := s;
    vp^.minut := m;
    plus := vp;
    dispose(vp);
end;
procedure Vreme.wr;
begin
    write(v^.sat);
    write(':');
    write(v^.minut);
end;
procedure Vreme.wrln;
begin
    wr; writeln;
end;
procedure Vreme.read;
var s, m : Int;
begin
    s.init; m.init;
    write('sat   '); s.read_limit(0,24);
    write('minut  '); m.read_limit(0,60);
    v^.sat := s.je;
    v^.minut := m.je;
    s.done; m.done;
end;
{ ------  pocetak glavnog programa ------ }
var z : Znak;
    b, b1, b2 : Float;
    i, i1, i2 : Int;
    v, v1, v2 : Vreme;
    vp        : vrPok;
    s         : word;
begin { velicina }
    ClrScr;
    writeln('TESTIRANJE KLASE Znak');
    z.init;
    z.postaje('z');
    z.wr;
    write(' ',z.manjeod ('u')); write('
',z.jednako ('x'));
    writeln(' ',z.izme"u ( 1 'u'));
    write(z.max ('D')); write(' ', z.min
('D'));
  writeln(' ',z.je);
    writeln('TESTIRANJE KLASE Float');
    b.init; b1.init; b2.init;
    b.postaje(3); b1.postaje(5.0);
b2.postaje(-1.5);
    write(b.manjeod(b1.je)); write('
',b.jednako(b.je));
    writeln(' ', b.izme"u(b1.je, b2.je));
    writeln('maksimum od ', b1.je, ' i ',
      b2.je, ' je ', b.max(b1.je));
    writeln('minimum od ', b1.je, ' i ',
      b2.je, ' je ',b.min(b2.je));
    writeln('zbir ', b.je, ' i ', b1.je, '
je ',
      b.plus(b1.je));
    b.read; b.read_limit(0.0, 10);
    write('eksponent broja ', b.je, ' je ',
b.exponent);
    b.done; b1.done; b2.done;
    writeln; writeln('TESTIRANJE KLASE Int');
    i.init; i1.init; i2.init;
    i.postaje(3); i1.postaje(5); i2.postaje(-
1);
    write(i.manjeod(i1.je)); write('
',i.jednako(i.je));
    writeln(' ', i.izme"u(i1.je, i2.je));
    writeln('maksimum od ', i.je, ' i ',
      i1.je, ' je ', i.max(i1.je));
    writeln('minimum od ', i.je, ' i ',
      i2.je, ' je ', i.min(i2.je));
```

```
      writeln(i.je, ' + ', i1.je, ' = ',
i.plus(i1.je));
      i read, i read_limit(3, 10);
      writeln('eksponent broja ', i.je, ' je
', i.exponent);
      i.done; i1.done; i2.done;
      writeln('TESTIRANJE KLASE Vreme');
      v.init; v1.init; v2.init;
      v.postaje(9,15); v1.postaje(24,55);
v2.postaje(3,6);
      write(v.manjeOd(v1.je)); write(' ',
v.jednako(v.je));
      writeln(' ', v.izmedju(v1.je, v2.je));
      vp := v.max(v1.je); v.postaje(vp^.sat,
vp^.minut);
   v.wr; write(' ');
      vp := v.min(v2.je); v.postaje(vp^.sat,
vp^.minut);
      v.wr; write(' ');
      v.wr; write(' + '); v1.wr; write(' = ');
      vp := v.plus(v1.je); v.postaje(vp^.sat,
vp^.minut); v.wrln;
      write('ucitavanje vremena '); x := wherex
      v.read; gotoxy(x + 22, wherey - 1); v.wr
      v.done; v1.done; v2.done;
end. { velicina }
```

# OBJEKAT Tekst

Svaki jezik ima svoje osnovne tipove – brojeve, slova, datoteke itd. Da bi se jezik mogao realizovati i naučiti broj osnovnih tipova i operacija nad njima je uvek mali, ali se zato ostavlja mogućnost proširivanja jezika. Sa objektnim *Turbo Pascal*-om programer može da napravi nove tipove podataka i to tako da budu (skoro) po svemu ravnopravni sa ugrađenim tipovima. Jedina prava razlika je na sintaksnom nivou, jer je moguće napisati sledeće

```
var A, B, C : integer;

C := A + B;
```

dok nije moguće napisati isto to za objekte

```
var A, B, C : Matrica;

C := A + B
```

iako bi to bilo sasvim prirodno (sabiranje matrica je u teoriji dobro definisano). U drugim objektnim jezicima (*C++*, *Smalltalk*) to je sasvim moguće i često se koristi. Međutim umesto znaka plus uvek se može navesti poziv procedure tako da nema stvarnog gubitka funkcionalnosti jezika. To znači da će se sabiranje svih objekata pisati npr. kao SAB(A,B) dok će sabiranje ugrađenih tipova ostati sa znakom +. To i bez ikakvih objektnih proširenja odavno postoji. Na primer „sabiranje" tekstova se potpuno ravnopravno piše i sa znakom + i pomoću standardne procedure **Concat**.

Drugim rečima, programi koji koriste nova tipove podataka pišu se uglavnom pozivanjem metoda. Takav programski tekst malo liči na uobičajenu matematički orijentisanu sintaksu algolskih jezika ali to je i jedina mana tog pristupa.

Programski primeri u ovom poglavlju značajno proširuju mogućnosti tipova kao što su **string**, **integer**, **real** a uvode i nove tipove za koje važe iste operacije – vreme, na primer.

Tekst nije u paskalu rešen kao samostalni tip nego kao niz znakova. Sledeći program bavi se tekstualnim promenljivama. Program **demoStr** definiše tri objekta: **Tekst**, **ObicanTekst** i **VelikiTekst**. **Tekst** je apstraktna klasa i služi za definisanje protokola za nasledničke klase. Svi njegovi metodi su prazni, nasledničke klase će ih napisati u skladu sa strukturom podataka koja predstavlja tekst.

**ObicanTekst** je skoro isto što i ugrađeni tip **string**. Jedina razlika je što mu se sada prilazi preko metoda a ne direktno. Ograničenje tog pristupa je isto kao i kod samog tipa string: tekst može biti dugačak samo do 255 znakova. To je u većini normalnih primena dovoljno ali za pisanje pravih korisničkih programa to je malo. Nije teško zamisliti da program treba da učita tekst sa celog ekrana i smesti u običan niz ali ne i u promenljivu tipa **string**. Otuda potreba za objektom **VelikiTekst**. On se sastoji iz celog broja u kome je dužina teksta i niza znakova. Taj niz može zapremati skoro svih 64K u jednom segmentu što se u praksi pokazalo kao dovoljno. **VelikiTekst** je za nekoliko metoda bogatiji od klase koju nasleđuje (**Tekst**).

Način kodiranja u objektu **VelikiTekst** bitno se razlikuje od prethodnih primera ali se zapravo radi o standardnoj tehnici. U objektu se i dalje nalazi samo jedna varijabla, s, ali ona više nije osnovnog tipa **dugiTekst**, već se deklariše kao pokazivač na **dugiTekst**, tj. **dugiTekstPok**. Ova promena je nužna da bi mogla postojati funkcija **je**, kojom se u glavnom programu dobija trenutna vrednost objekta. Funkcija ne može da vrati nijedan tip osim onih koji su zaista ugrađeni u *Turbo Pascal* (još jedna razlika!) ali može da vrati pokazivač na slog (ili objekat). To uslovljava značajne sintaksne promene unutar samog objekta i treba se na njih privići.

Pošto je s pokazivač, konstruktor mora da pozove standardnu funkciju **New** da bi kreirao s. Zato i done više ne postoji formalnosti radi, u njemu se nalazi odgovarajuća funkcija **Dispose**. Naravno, u svim metodima elementi sloga dosežu se navođenjem pokazivača, npr. s^ a i svi parametri metoda moraju biti tipa **dugiTekstPok**.

Nezavisno od toga, interesantna je primena naredbe **move** u većini metoda. Sva premeštanja bi se mogla obaviti i običnim for petljama ali **move** je brže.

Datoteka **DemoStr** sadrži i glavni program koji prikazuje osnovne primene klasa **ObicanTekst** i **VelikiTekst**. Postoji i mnogo drugih operacija koje bi se mogle izvoditi nad tekstom – poravnavanje ulevo i udesno, prikaz na određenom delu ekrana itd.

Može se, naravno, dogoditi da ni 64K nije dovoljno. Tada je najbolje slati suvišne znakove na disk, ali to podrazumeva sasvim drugi pristup tekstualnim objektima.

```
DemoStr; { str }
uses crt
type
    Tekst = object
      procedure init;
      procedure pridruzivanje;
      procedure umetanje;
      procedure brisanje;
      procedure pozicija;
      function duzina : integer;
      procedure kopiranje;
      procedure wr;
      procedure wrln;
      procedure postaje;
      function je : string;
      end;
{ ------------  metodi za Tekst ---           }
procedure Tekst.init;
begin
end; { tekst.init }
procedure Tekst.pridruzivanje;
begin
end; { tekst.pridruzivanje }
procedure Tekst.umetanje ;
begin
end; { tekst.umetanje }
procedure Tekst.brisanje;
```

```
begin
end  { tekst brisanje }
procedure Tekst pozicija;
begin
end; { tekst pozicija }
procedure Tekst kopiranje;
begin
end  { tekst kopiranje }
function Tekst duzina : integer
begin
end  { tekst duzina }
procedure Tekst wr;
begin
end  { Tekst write }
procedure Tekst wrln;
begin
end  { Tekst wrln }
function Tekst je
begin
end  { je }
procedure Tekst postaje
begin
end; { Tekst postaje }
{ --        -       - objekat ObicanTekst          }
type
    ObicanTekst = object(Tekst)
      s : string;
      constructor init
      destructor Done; virtual;
      procedure pridruzivanje(st1   string);
virtual;
      procedure umetanje(st2   string; p   in-
teger)  virtual
      procedure brisanje(p, len   integer);
virtual;
      function pozicija(st2   string)   inte-
ger; virtual;
      function duzina   integer;  virtual
      procedure kopiranje(st2 : string
          p, len ; integer); virtual
      procedure wr; virtual;
      procedure wrln; virtual
      procedure postaje(st : string)  virtual;
      function je   string; virtual
      end; { ObicanTekst }
{           metodi za ObicanTekst ---- ----   }
constructor ObicanTekst init
begin
     s  = ''
end  { tekst init }
destructor ObicanTekst Done;
begin
end  { ObicanTekst Done }
procedure ObicanTekst pridruzivanje;
begin
     s  = s + st1;
end  { ObicanTekst pridruzivanje }
procedure ObicanTekst umetanje
begin
     insert(st2, s, p);
end; { ObicanTekst umetanje }
procedure ObicanTekst brisanje
begin
     Delete(s,p len);
end; { ObicanTekst brisanje }
function ObicanTekst pozicija
begin
     pozicija := Pos(st2,s)
end  { ObicanTekst pozicija }
procedure ObicanTekst kopiranje;
begin
     s := Copy(st2,p len);
end  { ObicanTekst kopiranje }
function ObicanTekst duzina;
begin
     duzina := length(s);
end; { ObicanTekst duzina }
procedure ObicanTekst wr
begin
     write(s);
end  { ObicanTekst write }
procedure ObicanTekst wrln;
begin
     writeln(s);
end; { ObicanTekst wrln }
function ObicanTekst je;
begin
     je := s;
end; { ObicanTekst je }
procedure ObicanTekst postaje;
begin
     s := st,
end, { ObicanTekst postaje }
{ ------- - tipovi za VelikiTekst ---- ---- }
const maxduzina = 1500;
type
    dugiNiz = array [ 1  maxDuzina ] of char;
    dugiTekstPok = ^dugiTekst
    dugiTekst = record
              duzina : integer
              ch : dugiNiz;
              end;
{ -------- objekat VelikiTekst ----------- }
type
    VelikiTekst = object(Tekst)
      s: dugiTekstPok;
      constructor init
      destructor done; virtual;
      procedure pridruzivanje(st1 ;
dugiTekstPok); virtual
      procedure umetanje(st2 ; dugiTekstPok
            p : integer); virtual;
      procedure brisanje(p, len   integer)
virtual;
      function pozicija(st2 ; dugiTekstPok);
integer; virtual;
      function duzina   integer   virtual;
      procedure kopiranje(st2   dugiTekstPok
             p, len   integer)  virtual
      procedure wr; virtual;
      procedure wrln  virtual;
      procedure postaje(st : dugiTekstPok);
virtual
      function je : dugiTekstPok; virtual;
      procedure iz_obicnog_teksta(st2 :
string; p : integer)
      end  { VelikiTekst }
{            metodi za VelikiTekst             }
constructor VelikiTekst init
var i : integer
begin
     new(s)
     s^ duzina := 0
     for i  = 1 to maxDuzina do s^ ch[i]  =
chr(0);
end  { VelikiTekst init }
destructor VelikiTekst Done;
begin
     Dispose(s)
end  { VelikiTekst Done }
procedure VelikiTekst pridruzivanje,
begin
     move(st1^ ch[1], s^ ch[s^ duzina + 1],
st1^ duzina);
     s^ duzina := s^ duzina + st1^ duzina
end  { VelikiTekst pridruzivanje }
procedure VelikiTekst umetanje
var st3 : dugiTekst;
begin
  move(s^ ch[1], st3 ch[1]  p   1);
  move(st2^ ch[1], st3 ch[p], st2^ duzina);
  move(s^.ch[p], st3.ch[p + st2^.Duzina]
```

```
(st2^ Duzina - p) + 1)
  st3 Duzina := st2^ duzina + s^ Duzina
  if st3 Duzina > maxduzina then st3 Duzina :=
maxduzina;
  s^ := st3
end; { VelikiTekst.umetanje }
procedure VelikiTekst.brisanje
var st2 : dugiTekst;
begin
  move(s^ ch[1], st2 ch[1], p - 1)
  move(s^ ch[(p + len)], st2 ch[p], (s^ duzina -
  (p + len)) +1)
  st2 Duzina := (s^ duzina - len);
  s^ duzina := st2 duzina;
  move(st2 ch[1], s^ ch[1], st2 duzina);
end; { VelikiTekst.brisanje }
function VelikiTekst.pozicija
var postoji : boolean
    i, j, kraj : integer;
begin
     kraj := abs(st2^ duzina - s^ Duzina) + 1;
     for i := 1 to kraj do begin
         postoji := true
         j := 1
         repeat
               if s^ ch[i + j - 1] <>
st2^ ch[j] then
                  postoji := false
               j := j + 1
         until (not postoji) or (j = st2^ duzina)
         if postoji then begin
            pozicija := i;
            exit;
            end
         end
     pozicija := 0;
end { VelikiTekst.pozicija }
procedure VelikiTekst.kopiranje
begin
     move(st2^ ch[p], s^ ch[1], len);
     s^.Duzina := len;
end; { VelikiTekst.kopiranje }
function VelikiTekst.duzina;
begin
     duzina := s^ Duzina
end; { VelikiTekst.duzina }
procedure VelikiTekst.wr
var i : integer;
begin
     for i := 1 to s^ duzina do wri-
te(s^ ch[i]).
end; { VelikiTekst.write }
procedure VelikiTekst.wrln
var i : integer;
begin
     for i := 1 to s^ duzina do wri
te(s^ ch[i]); writeln;
end; { VelikiTekst.wrln }
function VelikiTekst.je;
begin
     je := s
end; { VelikiTekst.je }
procedure VelikiTekst.postaje
begin
     move(st^ ch[1], s^ ch[1], st^ duzina);
     s^ duzina := st^ duzina;
end; { VelikiTekst.postaje }
procedure VelikiTekst.iz_obicnog_teksta
var d : integer;
begin
     d := length(st2)
     move(st2[1], s^ ch[p], d);
     if (p + d) > s^.duzina then s^.duzina :=
p + d - 1
end; { VelikiTekst.iz_obicnog_teksta }
{ --- verijable za Glavni Program (demo) -- }
var t : string;
    t1, t2 : ObicanTekst;
    t3, t4 : VelikiTekst;
    t5     : dugiTekstPok
    i, b : integer;
    t6 : dugiNiz;
begin { demoStr }
      clrscr;
      t := 'Ovo je nulta proba za tekst ';
      write(t); writeln;
      t1 init;
      t1.postaje('Druga proba za ObicanTekst');
      t1.wrln;
      writeln('Duzina gornjeg teksta je ',
t1 duzina);
      t := t1 je;
      writeln(t,'///////////////');
      t1 pridruzivanje('[[[[[[[[[(');
      t2 init  t2.postaje('abcdefghijkl');
      t2.umetanje('so', 4);
      writeln(t2.je),
      t2.brisanje(4, 3);t2 wrln
      writeln(t2 pozicija('jkl'))
      t2 kopiranje('qwerrtuyi',2 4);
      t2 wrln;
      t3 init; t4 init
      t3 iz_obicnog_teksta('PRVI PRIMER ZA Ve-
liki Tekst',1)
    t3 wrln;
      t3 iz_obicnog_teksta('nastavak ele-
ai',20); t3 wrln;
      t5 := t3 je;
      b := t5^ duzina;
      t6 := t3 je^ ch;
      for i := 1 to b do write(t6[i]);writeln;
      t3 iz_obicnog_teksta('Drugi primer za Ve-
liki Tekst',1);
      t4 postaje(t3 je);t4 wrln;
      t4 iz_obicnog_tek
sta('1234567890123456789',1);
      write('t4 je ') t4 wrln
      t3 kopiranje(t4 je,20,14);write('t3 je
');t3 wrln;
      writeln('t3 se sadrzi u t4 i pocinje na
mestu ',
        t4 pozicija(t3 je));
      t3 brisanje(1,4) writeln('t3 posle brisa-
nje ') t3 wrln;
      t4 done  t4 init
      t4 iz_obicnog_tek
sta('123456789012345678901234567890',1)
      t3 umetanje(t4.je, 9);writeln('t3 posle
umetanje ');t3 wrln;
      t4 brisanje(1,t4 duzina);
      t4 iz_obicnog_teksta('qazwsxedcrfvtgby-
hnujmik,ol.p,/',1)
    t4 wrln,
      t3 brisanje(1,t3 duzina)
      t3.iz_obicnog_teksta('ssasssssasaasssssss-
ssssss',1)
    t3 wrln;
      t3.pridruzivanje(t4.je);t3.wrln
      t1 done  t2 done; t3 done; t4 done
writeln('tip za t1 ', typeof(t4)=typeof(t3));
end  { demoStr }
```